SONDERHEFT 83 Markt&Technik öS 120,-/sfr 16,-/Lit 16000 hfl 21,-/dkr 75,-/fmk 62,- DM 16,-



FLOPPY

Magische Spuren

95 Blocks mehr auf Disk!

Auferstehung

Die 790-KByte-Floppy 1581 ist wieder da!

Super-Komfort

C 64-Tools: ersetzt 10 DOS-Utilities

Daten-Blitztransfer

4 Kopier-Tools für komplette Disketten oder Einzeldateien

über 40 Programme auf Diskette

64'er

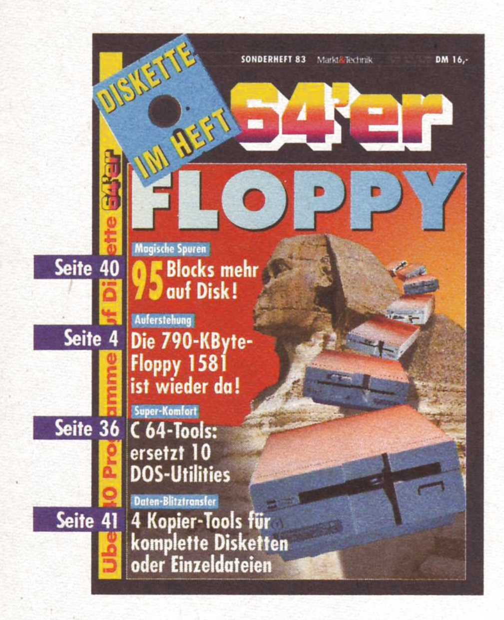
Die neue AMIGA POWER DISK NR.13: Lassen Sie sich dreidimensional animieren - für nur 19,80 DM!

Exklusiv auf der POWER DISK NR.13: Die vollständige Version von "Amiga 3D-Sprinter V1.3"!
Das konnten bisher nur teure Raytracing-Programme: Spiegelungen und Schatten in Echtzeit! Das Animations-Programm "Amiga 3D-Sprinter V1.3" berechnet dreidimensionale Animationen in Sekundenschnelle - natürlich in Farbe. Dabei ist es einfach zu bedienen und eröffnet auch Einsteigern die faszinierende Welt der Computeranimation. Zusätzlich zu zahlreichen Beispielen ist auch eine Turbo-Version für Beschleunigerkarten in diesem einmaligen Paket enthalten. Und wenn Sie Freunden Ihre Werke präsentieren wollen - kein Problem: Mit dem separaten Abspielprogramm haben Sie's auf Video!
Steigen Sie ein - nie war es günstiger, die dritte Dimension zu erreichen!

Nutzen Sie dieses Angebot zum Wahnsinns-Preis von nur 19,80 DM!

Ab sofort bei Ihrem Zeitschriftenhändler!





Alle Programme aus Artikeln mit einem ▪-Symbol finden Sie auf der beiliegenden Diskette (Seite 19)



Frohe Kunde für alle C-64- und C-128-Besitzer: Ab sofort gibt's die Commodore-Super-Floppy 1581 wieder zu kaufen. Wir haben das Laufwerk unter die Lupe genommen.

Die Kleinste ist die Größte! Was den IBM-kompatiblen ATs, dem Amiga, Atari ST oder Archimedes ohnehin als Selbstverständlichkeit erscheint, ist auch dem C 64 recht und billig: Datenspeicherung auf 3½-Zoll-Disketten! Diese Micro-Disketten haben gegenüber den 5¼-Floppy-Disks der Laufwerke 1541 oder 1571 immense Vorteile:

- Die empfindliche Magnetscheibe sitzt in einem stabilen Kunststoffgehäuse (Knicken und Verbiegen z.B. beim Transport ausgeschlossen!),
- mechanischer Schreibschutz per Mini-Plastikschieber in der rechten oberen Ecke (Abb. 5), den man nach Bedarf einsetzt (Schreibschutzaufkleber entfallen ersatzlos),
- die empfindliche Oberfläche der Magnetscheibe ist durch einen Metallschieber geschützt.

Die 1581 wurde bereits 1987 einige Monate von Commodore Deutschland vertrieben und verschwand sang- und klanglos vom Markt. Begründung: Inkompatibilität mit vielen kommerziellen Software-Produkten (vor allem kopiergeschützte), die für andere C-64-Laufwerke (1541, 1551, 1570, 1571) ungewöhnliche Diskettengröße – und, weil die Software-Hersteller nicht mitspielten (3½-Zoll-Disketten kosteten damals viermal so viel wie 5¼-Zoll-Scheiben). Vielleicht war Commodore auch schlecht beraten, die kleine Super-Floppystation speziell C-128-Besitzern als Ersatz für die 1571 oder zumindest als Zweitlaufwerk anzubieten. So lobenswert diese Werbestrategie für die seit jeher sträflich vernachlässigten C-128-Anwender auch war – wo blieben die C-64-Besitzer? Das Argument C 128 zieht nämlich nur, wenn man die Floppy in der Betriebsart des BCIS (Burst Command Instruction Set) verwendet: Der Burst-Modus schaltet das Laufwerk auf schnelle serielle Datenübertragung um und besitzt einen erweiteten Floppy-Befehlssatz. Das BCIS ist auch in den 1570- und 1571-Diskstationen enthalten.

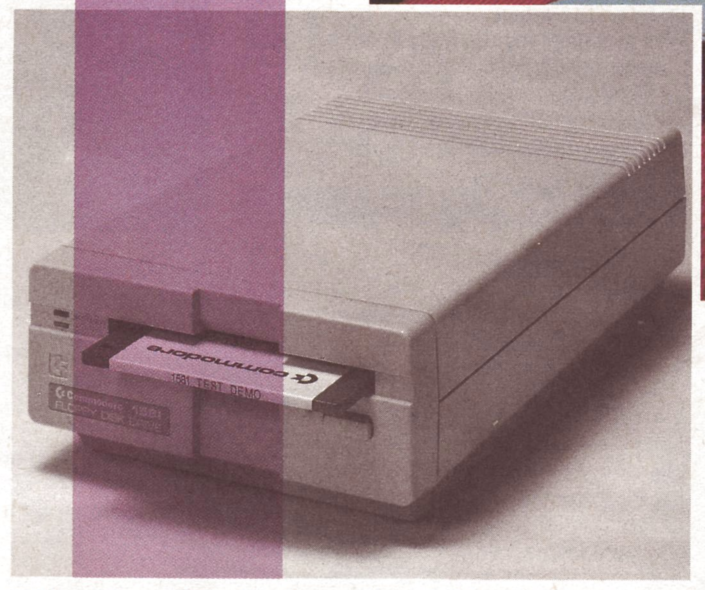
Quasi als 1541 am C 64 angeschlossen erledigt die 1581 alle Aufgaben ebenfalls zur vollsten Zufriedenheit des Anwenders – aber mit fast dreifacher Speicherkapazität pro Disk und um 50 Prozent schneller! Häufiger Diskettenwechsel ist out!

Vorteile der 3½-Zoll-Micro-Disketten

Als man das Laufwerk Anfang 1988 aus dem Verkehr zog, hat man dabei Tausende von C-64-Besitzern vergessen, die das Laufwerk vor allem als Zweitfloppy speziell zur Datenspeicherung verwenden wollten (z.B. Textverarbeitungsdateien oder Dateiverwaltungs-Files): Eine 3½-Zoll-Disk bietet immerhin 790 KByte Speicherkapazität: 808 960 Byte (fünfmal so viel wie eine einseitige 5¼-Zoll-Disk der 1541!). Nach dem Formatieren zeigt das Directory »3160 Blocks free«.

3½-Zoll-Disketten lassen sich nur doppelseitig verwenden. Schreibkerbe knipsen, Diskette umdrehen um auf die Rückseite zu speichern (Speicherplatz horten wie bei der 1541), ist bei der Micro-Disk unmöglich und unnötig: Beim Formatieren werden 80 Spuren mit 40 logischen Sektoren 256 Byte eingerichtet. Physikalisch sind's nur 20 Blocks pro Spur, denn die 1581 erzeugt grundsätzlich 512-Byte-Sektoren: intern werden sie aber vom Floppy-DOS in zwei logische Datenblöcke 256 Byte eingeteilt (sonst wäre die Kompatibilität zum C 64 und 99 Prozent aller Software nicht mehr gewährleistet). Der gravierende Unterschied zur 1541: Die Systemspur für BAM und Directory wird nicht mehr in Track 18, sondern in Spur 40 angelegt. Deshalb haben selbstverständlich Backup-Programme oder Disk-Tools, die auf die Besonderheiten der 1541 zugeschnitten sind, bei der 1581 keine Chance.

Die Floppy-Verwaltung (DOS Version 10.0) unterscheidet sich ebenfalls erheblich vom DOS 2.6 der 1541. Das neue Betriebssystem versteht die alten Befehle und kann einiges mehr: Man kann v»Partitions« (Unterverzeichnisse) im Directory einrichten. Die Disk läßt sich in verschiedene Bereiche aufteilen – es müssen nicht alle Dateien wie bei der 1541 im



[1] Micro-Floppy 1581 mit externem Netzteil: 6,3 x 14 x 23 cm.

Hauptverzeichnis (Root-Directory) abgelegt werden. Alle Partitionen erhalten als Kennzeichen einen bislang unbekannten File-Typ: CBM. Diese komfortable Funktion erinnert sehr an die Fähigkeiten z.B. der MS-DOS-, Amiga und Archimedes-Rechner, Subdirectories zu erzeugen.

Hardware identisch

Die neue 1581 unterscheidet sich äußerlich überhaupt nicht von der alten Version: Sie wiegt anderthalb Kilo und mißt nur 14 x 23 cm (Abb. 1). Das Netzteil muß nach wie vor extern angebracht werden. Neben den beiden seriellen Anschlüssen



bel mit den Versionen 3.0/3.1.

Als Dateitypen kennt die 1581 ebenfalls DEL, SEQ, PRG, USR und REL. An Struktur und Bedienung solcher Files hat sich gegenüber der 1541 nichts geändert. Die Diskanweisungen NEW (formatieren), VERIFY, SCRATCH, RENAME, COPY und VALIDATE arbeiten wie gewohnt, ebenso LOAD und SAVE des Basic 2.0. Interessanter wird's bei den floppyinternen Befehlen (wie B-R, B-W, M-R, M-W usw.): Neu ist das PARTITION-Kommando, das sich nur bei geöffnetem Befehlskanal eingeben läßt. Es generiert separate Unterverzeichnisse auf der Disk. Dieser Speicherplatz muß mindestens 120 Sektoren groß sein (also drei Spuren 40 Blocks) und darf auf gar keinen Fall die Systemspur 40 enthalten! Dazu gibt's ein Demoprogramm auf der Diskette zu diesem Sonderheft:

```
LOAD "SUBDIR.1581",8
```

Die 1581 darf mit beliebiger Geräteadresse (Nummer des Laufwerks, von 8 bis 11) angeschlossen sein. Vor dem Start mit RUN sollten Sie einen Floppyreset im Direktmodus ausführen:

```
OPEN 1,(gewählte Geräte-
nr.),15,"UJ": CLOSE 1
```

Geben Sie nach dem Start mit RUN Gerätenummer, Diskettenname und ID ein. Auf Tastendruck wird die 3½-Zoll-Disk zunächst neu formatiert, dann erscheinen die Fragen nach Subdirectory-Namen, Anfangsspur, Anfangssektor (normal: 0) und Sektoranzahl (sie muß immer durch »40« teilbar sein. Anzahl der verwendeten Spuren also = Sektormenge : 40).

Ein Beispiel: Ab Spur 30 wollen Sie 240 Sektoren als Subdirectory einrichten. Dann müssen Sie diese Eingaben erledigen, wenn das Programm danach fragt:

```
Anfangsspur: 30
Anfangssektor: 0
Wieviele Sektoren: 240
```

Der Tipp auf <RETURN> veranlaßt die Floppy 1581, eine Partition einzurichten. Die Spuren werden erneut formatiert und mit dem angegebenen Unterverzeichnisnamen als Subdirectory eingerichtet. Dieses Prinzip verfolgt z.B. auch unser Programm »Ultraformat V2« auf derselben Sonderheftdiskette für die Spuren 36 bis 40 einer 5¼-Zoll-Disk. Am File-Typ CBM ist die Partition im Hauptdirectory zu erkennen. Näheres steht im Handbuch zur 1581, S. 86.

Der nächste Unterschied zum DOS 2.6 ist der Burst-Modus (schnelle serielle Datenübertragung, wie bei der 1571). Machen Sie sich keine falschen Hoffnungen: der Modus ist nur mit dem C 128 zu erreichen, da dieser die Hard- und Software-Voraussetzungen des Betriebssystems mitbringt. Im Burst-Modus arbeitet die Floppy mit einer Übertragungsrate von im-

(Serial und Interface, zwei 6polige DIN-Buchsen, identisch mit den 1541/1571) auf der Gehäuserückseite findet man die DIP-Schalter zur Einstellung der Geräteadressen (8 bis 11), den Netzteilanschluß (Power) und den Netzschalter.

Der interne Aufbau der 1581 (Abb. 2) ist von dem der 1541 grundverschieden. Gemeinsam haben beide Floppytypen nur den Mikroprozessor 6502. Daneben wird die 1581 vom IBM-PC-kompatiblen Disk-Controller WD 1772 gesteuert, außerdem stellt es 32 KByte ROM und einen RAM-Speicher von 8 KByte (RAM 4364) zur Verfügung (die 1541 z.B. hat nur 2 KByte!) und benutzt den CIA-Chip 8520A (s. Tabelle). Die 1581 arbeitet nach dem Prinzip eines Caché-Speichers: Es wird immer der Dateninhalt einer **kompletten Spur** in die internen Puffer übernommen (nicht nur sektorweise). Ein Riesen-Plus für die Arbeitsgeschwindigkeit!

Neues Betriebssystem: DOS V 10.0

Neben der Hardware mußte man natürlich auch ein komplett neues Betriebssystem entwickeln, um die immense Speicherkapazität und neue Funktionen der 1581 zu verwalten: das Floppy-DOS mit der Versionsnummer 10.0 (Vergleich: 1541 = DOS 2.6, 1571 = DOS 3.0 bzw. 3.1 im C-128D-Blech). Grob betrachtet ist das 1581-DOS fast völlig abwärtskompati-

merhin 1 Million Bit/s (die 1541 schafft dagegen nur 2400 Bit/s). Die Burst-Routinen funktionieren bei der 1581 aber um einiges schneller als bei der 1571: durch den erwähnten Caché-Speicher!

So läuft's mit den bislang bekannten Commodore-Laufwerken 1541 und 1571: Beim Laden eines Programms oder einer Datei (also mehrere Sektoren), wird der erste zunächst in einem internen Floppypuffer zwischengespeichert, dann kommt die Übertragung zum Computer und der logisch nächste Sektor wird geholt. Die 1581 macht's anders: Sie packt eine gesamte Spur mit 40 Sektoren ins Caché-RAM – und arbeitet damit doppelt bis dreimal so schnell wie die ohnehin nicht langsame 1571! Gegenüber der 1541 bedeutet das sogar 30fache Geschwindigkeit (nur im Burst-Modus!). In der 1541-Betriebsart ist die 1581 30 Prozent schneller.

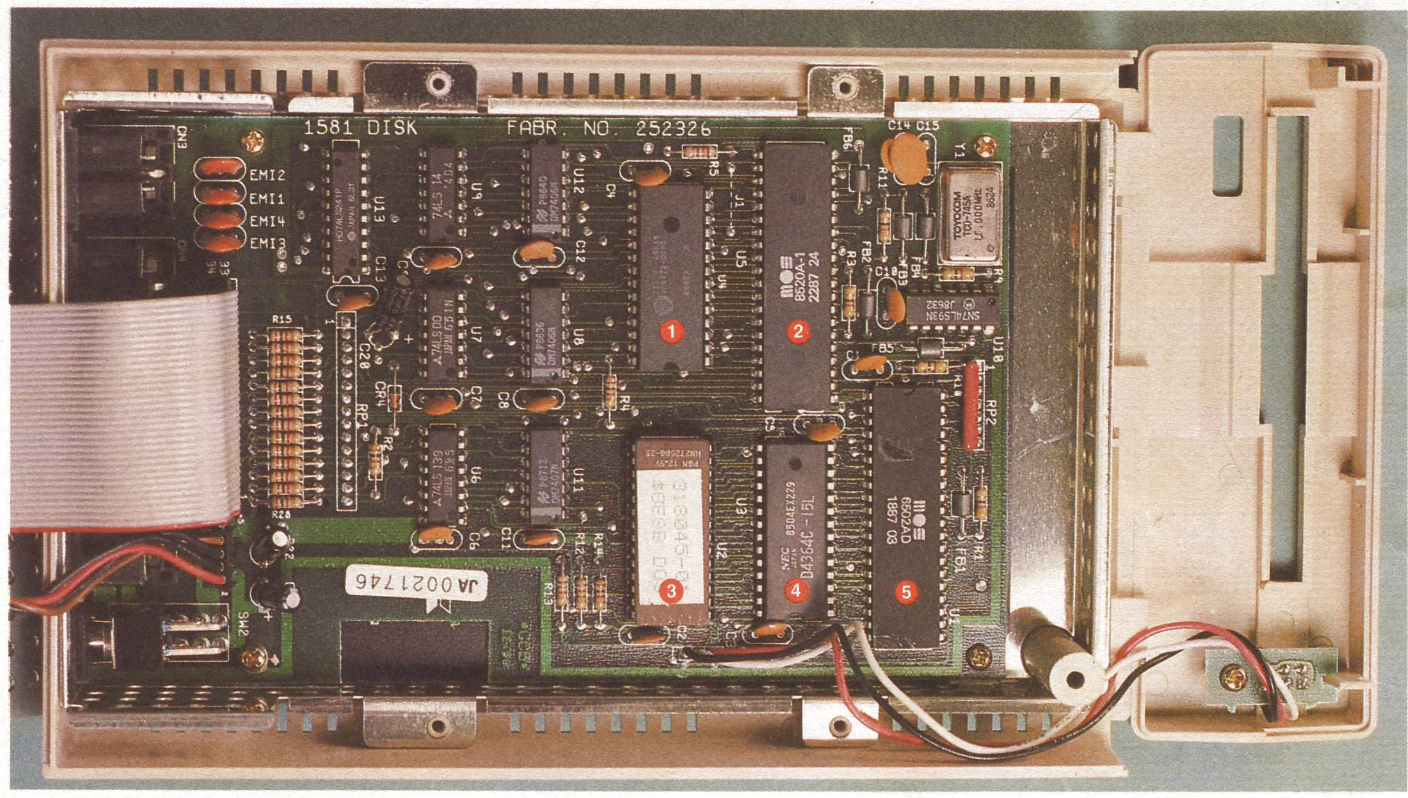
Wie beim PC: Caché-RAM

Prüfen wir nach, wie der Caché-Speicher funktioniert:

```
LOAD "PRG 1.BAS",8
```

Achtung: Falls Sie eine andere Geräteadresse als »8« benutzen, müssen Sie das in Zeile 10 berichtigen, bevor Sie RUN eingeben!

Das Demoprogramm veranlaßt die 1581 nach dem Start, Spur 1 ins Caché-RAM zu laden. Wenn der Laufwerksmotor stoppt, muß man eine beliebige Taste drücken: Der Bildschirm meldet sich wieder mit READY. Obwohl in Zeile 50 der



[2] Die Platine der 1581: Die Zahlen bedeuten:
1 – Disk-Controller WD 1772
2 – CIA 8502A
3 – 32 KByte-Betriebssystem-EPROM
4 – RAM 4364 (8 KByte)
5 – Mikroprozessor 6502A

erneute Ladebefehl für den Caché-Speicher steht, verharrt die 1581 regunglos: Spur 1 befindet sich ja bereits im Caché-RAM. Es liegt hier ab Adresse $0C00 (3072) bis $1FFF (8191) im Floppyspeicher. Wenn man dessen Lage kennt, ist es ein Kinderspiel, die Daten mit M-R auszulesen.

Das beweisen die anderen Demo-Files auf unserer Sonderheftdiskette:

```
LOAD "PRG 2.BAS",8
```

Achtung: Hier ist die Geräteadresse in Zeile 10 schon auf »9« eingestellt.

Das Beispielprogramm überträgt eine Zahlenreihe in die Spur 79: Nr. 0 bis 19 als erstes Byte aller 20 möglichen Sektoren (also die Zahl 0 in Sektor 0, 1 in Sektor 1 usw.). Um diese 20 Datenblöcke wieder zu lesen, müssen Sie unser drittes Demoprogramm laden:

```
LOAD "PRG 3.BAS",8
```

Nach dem Start mit RUN bringt das Floppy-DOS die 20 Spuren von Spur 79 in den Caché-Speicher und liest nun die Bytes (M-R), die vorher per M-W in die Sektoren eingetragen wurden. Die Bildschirmausgabe geschieht in 256er-Schritten ab Pufferadresse 3072.

Da die 1581 nur ein einziges Diskettenformat verwendet (MFM, Modified Frequenzy Modulation), entfallen z.B. auch

Logisches und physikalisches Format

die nervtötenden Zugriffsversuche der Single-Floppy 1571 (DOS 3.0), wenn Sie eine einseitig formatierte Disk ins Laufwerk schieben und z.B. das Directory laden wollen: Die 1581 verwaltet die Daten der Directory-Spur ebenfalls im Caché-Speicher!

Alle anderen Commodore-Laufwerke (1541, 1551, 1570 und 1571) arbeiten mit Sektoren zu je 256 Byte und **einer** logischen Diskettenseite. Das gilt auch für die 1571, die zwar doppelseitig schreiben und lesen kann, beide Diskseiten aber wie eine einzige behandelt: Statt zwischen Seite 1 und 2 zu unterscheiden, werden einfach die Sektoren der zweiten Seite von 36 bis 70 durchnumeriert. Ähnlich, aber komplizierter, läuft's bei der 1581 ab. Physikalischer Aufbau einer 3½-Zoll-Disk:
- Sie besteht aus zwei Seiten,
- jede Seite hat 80 Spuren,
- jede Spur besteht aus zehn Sektoren,
- jeder Sektor umfaßt 512 Byte.

Jetzt kommt der Begriff »Zylinder« ins Spiel, den manche Computer-Freaks aus der MS-DOS-Welt kennen. Ein Zylinder besteht aus den äquivalenten Spuren beider Seiten:
- Die 3½-Zoll-Disk hat 80 Zylinder,
- ein Zylinder umfaßt 20 Sektoren (10 auf jeder Seite),
- der Sektor besteht aus 512 Byte.

Um die ROM-Routinen des C64 und der 1581 im C-64-Betrieb nicht durcheinanderzubringen, muß man den Sektorinhalt teilen, um Datenblocks à 256 Byte zu erhalten. Dann ergibt sich das endgültige logische Format:
- Eine Diskette hat 80 Zylinder (der Einfachheit halber werden sie ab sofort als einzelne Spuren bezeichnet, obwohl es eigentlich zwei sind!),
- ein Zylinder (also jetzt: Spur) besteht aus 40 Sektoren à 256 Byte.

Bei jedem Datenzugriff (Laden) bringt der Disk-Controller die physikalischen Datenblocks ins Floppy-RAM der 1581, das DOS 10.0 konvertiert sie aber automatisch ins logische Format und schickt sie erst dann zum C64:

Diskette (physikalisches Format) ->Controller ->DOS (logisch) ->Computer.

Beim Schreiben geht's genau umgekehrt.

Die logischen Spuren einer von der 1581 formatierten 3½-Zoll-Diskette werden von 1 bis 80, die Sektoren von 0 bis 39 numeriert. Der Burst-Modus ist im Kapitel 9, S. 101 im 1581-Handbuch mit allen signifikaten Bitbelegungen der Burst-Anweisungen ebenso gut (besser: schlecht) beschrieben wie in den 1571-Handbüchern. Eine Zusatzanweisung ist dazu gekommen: U0 + CHR$(29) = Dump Track Cach Buffer (Inhalt des Caché-RAM lesen). Diesen zusätzlichen Floppyspeicher kennen nämlich weder 1570 noch 1571.

```
                 Sector Editor

00 28 04 85 27 00 50 49 43 00 (..´.pic
08 2e 44 49 52 a0 a0 a0 a0 08 .dir....
10 a0 a0 a0 a0 a0 00 00 00 10 ........
18 00 00 00 00 00 00 00 00 18 ........
20 00 00 82 27 01 48 4f 57 20 ...´.how
28 20 54 4f 20 55 53 45 a0 28  to use.
30 a0 a0 a0 a0 a0 00 00 00 30 ........
38 00 00 00 00 00 00 32 00 38 ......2.
40 00 00 82 29 00 42 41 43 40 ...).bac
48 4b 55 50 31 32 38 2d 31 48 kup128-1
50 35 38 31 a0 a0 00 00 00 50 581.....
58 00 00 00 00 00 00 12 00 58 ........
60 00 00 82 29 12 42 41 43 60 ...).bac
68 4b 55 50 20 36 34 2d 31 68 kup 64-1
70 35 38 31 a0 a0 00 00 00 70 581.....
78 00 00 00 00 00 00 12 00 78 ........

Track: 40 ($28)   Drive:1581   Mode:MFM
Sector:03 ($03)   Unit: 8-D0

Status: 00, ok,00,00
```

[3] »Sector Editor« (Test/Demo-Disk): passabler Diskettenmonitor

Wer mehr über BCIS wissen möchte, dem empfehlen wir unser 128er-Sonderheft 82. Die darin veröffentlichten Informationen gelten fast unverändert auch für die 1581. Beispielprogramme muß man aber auf die neue Spur- und Sektorenverteilung der Micro-Floppy anpassen.

1581-Software auf der Test/Demo-Disk

Kommerzielle Software (Kopierprogramme, Disk-Tools und -Utilities) konnte sich wegen der veränderten Diskettenverwaltung schlecht auf die 1581 einstellen. Freaks und Hobby-Programmierer, die oft bessere Software zustandebringen als hochgelobte Softwarehäuser, hatten zu wenig Zeit, die 1581 ausreichend auszutesten: Sie wurde zu früh wieder vom Markt genommen. Dem Käufer stehen daher zunächst – außer den Programmen, die er selbst entwickelt – Dateien auf der Test/Demo-Disk zur Verfügung, die man gewohnheitsmäßig beim Kauf einer neuen Commodore-Floppy als Gratiszugabe bekommt. Was taugen die Programme auf der Gratisdiskette? Berücksichtigen sie alle Besonderheiten einer 3½-Zoll-Scheibe?

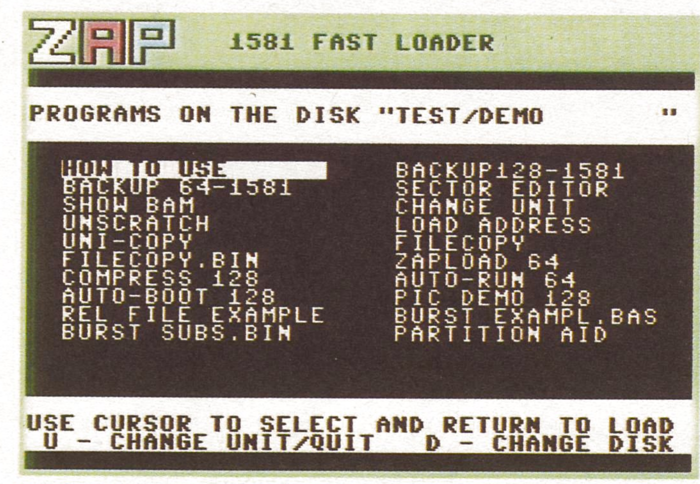
Backup 128-1581: spezielle C-128-Software. Zieht 1:1-Kopien einer 3½-Zoll-Diskette. Kopierdauer: acht Minuten. Bei der Quelldiskette sollte man auf alle Fälle den Schreibschutz aktivieren (Plastikschieber rechts oben öffnen!).

Backup 64-1581: macht Disketten-Backups mit dem C64. Das kann zehn Minuten dauern, bei bis aufs letzte Byte gefüllten 3½-Zoll-Disks noch länger. Mit weniger als 100 Dateieinträgen im Directory der Quelldisk sollte man ein File-Copy verwenden.

Sector Editor: ein Disketten-Monitor für die 1581, mit dem sich einzelne Datenblöcke laden, modifizieren und auf Diskette zurückschreiben lassen. Die gewünschte Geräteadresse kann man einstellen (<U> = Toggle Unit). Das Programm beginnt automatisch bei Spur 1, Sektor 0; man kann aber höhere Werte per Funktionstasten einstellen. Der Sektor-Scan wird mit <R> aktiviert und zeigt den Inhalt des gewünschten Datenblocks als Hex-Dump (Abb. 3). Per Taste <E> lassen sich die Hex-Bytes überschreiben (Bildschirm-Scrollen mit <CRSR aufwärts/abwärts>) und mit <W> geänderte Datensektoren wieder auf Disk zurückschreiben. Den BAM-Sektor finden Sie in Spur 40, Sektor 0. Mit »Sector Editor« lassen sich z.B. Disk-Name und ID-Kennung komfortabel ändern. Der erste Directory-Block beginnt normal in Spur 40, Sektor 3.

Show BAM: grafische Übersicht der Blockbelegung der aktuellen Diskette im Laufwerk (Kreis = Sektor belegt, Punkt = frei). Fehlerhafte Blocks werden nicht identifiziert. Offensichtlich ist das Utility für die 1581 nicht geeignet: Die BAM wird nämlich in Spur 18, Sektor 0 gesucht. Das hat bei den anderen Floppies seine Berechtigung, nicht aber bei der 1581: Dort müßte das Programm nach Sektor 0 in Spur 40 forschen.

Partition Aid: Die Hinweisdatei auf der Test/Demo-Disk meldet zu diesem Programm: Damit kann man die erwähnten Unterverzeichnisse (Partitions, mindestens 120 Sektoren) auf einer 3½-Zoll-Disk reservieren lassen und auf Wunsch dort ein Directory eintragen. Allerdings: Alle Ladeversuche blieben vergeblich – es war kein Programm im Computer (weder im C64 noch C128!). Schleunigst wurde mit »Sector Editor« der im Directory gezeigte Startblock (Nr. 1 in Spur 47) in den Computer geholt: 254 Null-Bytes grinsten uns entgegen...



[4] »Zapload 64«: Lademenü für aktuelle Disk im Laufwerk

Change Unit: Utility zur Umstellung der Geräteadressen von mehreren angeschlossenen Floppies.

Unscratch: stellt gelöschte Dateien wieder her, wenn die Disk in der Zwischenzeit nicht erneut beschrieben wurde.

Load Adress: nützliches Utility, um die Ladeadressen von Programmen oder Dateien im ersten Datenblock zu ändern. Interessant z.B. für Grafiken, die man zwischen zwei Malprogrammen mit unterschiedlichen Grafikspeicher-Startadressen austauschen möchte!

Uni-Copy: universelles File-Copy-Programm für den Datenaustausch zwischen **zwei** beliebigen Diskettenstationen, also nicht speziell für die 1581: Jede erdenkliche Variation ist

möglich (1541 - 1581 und umgekehrt, 1571 - 1541, 1581 - 1571 usw.). Damit lassen sich z.B. die Files der 1581-3½-Zoll-Test/Demo-Disk auf eine 5¼-Zoll-Scheibe in der 1541 oder 1571 übertragen (oder umgekehrt!).

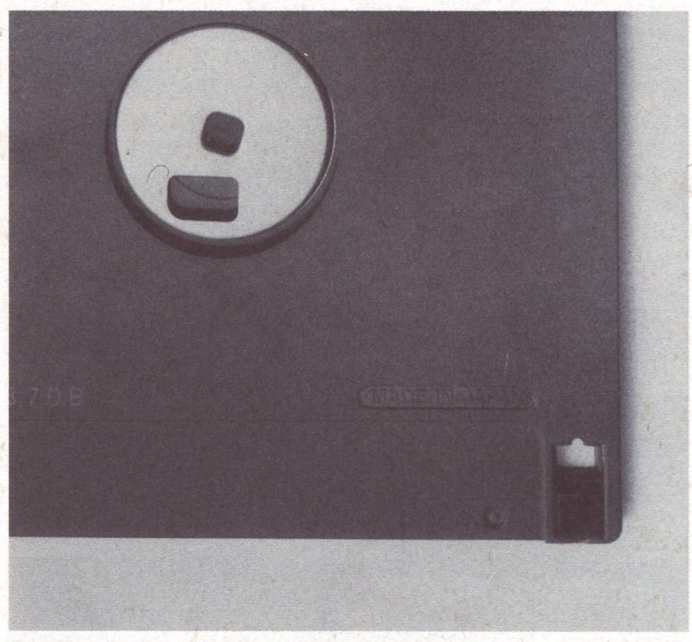
File-Copy: ein weiteres File-Copy-Tool für **zwei** Floppystationen. Es kopiert zusätzlich von Diskette in die RAM-Disk (z.B. 1764 oder 1750!).

Zapload-64: Lademenü mit eingebautem Schnellader (Abb. 4), universell lauffähig mit den Floppytypen 1541, 1571 und 1581. Auch andere Geräteadressen als 8 lassen sich einstellen.

Compress 128: waschechtes C-128-Programm, das nur mit diesem Computer läuft. Es komprimiert Programme und Dateien (Packer).

Auto-Run 64: Maschinensprache-Utility, das normale Assembler- und Basic-Programme mit einem Autostart ausstattet, der unmittelbar nach dem Laden aktiviert wird. SYS oder RUN können ab sofort entfallen.

Auto-Boot 128: erzeugt einen Startsektor (Spur 1, Sektor 0) und legt darin ein kurzes Maschinenprogramm ab. Damit läßt sich eine Diskette beim Einschalten oder per Druck auf den Reset-Knopf booten (automatisch starten, Basic 7.0-Befehl BOOT).



[5] Schreibschutzriegel bei 3½-Zoll-Disketten

Pic Demo 128: sollte eigentlich Hires-Grafiken aus der Partition »Pic.Dir« der Test/Demo-Disk laden und sie als Dia-Show zeigen (funktioniert nur beim C 128, da der Burst-Modus verwendet wird). Nur: Auf der Test/Demo-Disk unserer Test-Floppy 1581 ist die Partition Pic.Dir zwar als File-Typ CBM eingetragen - besteht aber aus 0 Blocks, ist also leer! Keine Spur von einem Ladeprogramm oder den Grafiken, die sich lt. Hauptprogramm im Subdirectory befinden müßten! Nach dem Start mit RUN bricht das Programm mit einer Fehlermeldung ab. Kopierfehler bei der Duplizierung der Test/Demo-Disk?

REL-File Example: Beispielprogramm zur Programmierung einer relativen Datei. Mehr über diesen speziellen Dateiverwaltungstyp findet man im Floppyhandbuch und in diesem Sonderheft.

Burst Example Files: Backup-Programm für zwei 1581-Laufwerke, das den Inhalt einer Quelldiskette Sektor für Sektor zur Zieldiskette überträgt: Die 3½-Zoll-Disk wird quasi dupliziert. Klappt nur mit dem C 128, weil Burst-Routinen (z.B. READ und WRITE) verwendet werden. »Burst Subs.bin« ist das Assembler-Programm, das nach dem Programmstart nachgeladen wird. Die Dateien »Burst Subs.scr« und »Burst Subs.lst« enthalten den Assembler-Quellcode und entsprechende Infos für Maschinensprache-Profis, die das Programm ändern oder eigene Programme entwickeln möchten, die den Burst-Modus der Floppy 1581 aufrufen. Vermutlich wird dieses Programm wenigen Anwendern etwas nützen, da kaum jemand zwei 1581er als Zweit- und Drittlaufwerke besitzt! Bislang geht's beim C 64 noch nicht ohne 5¼-Zoll-Hauptlaufwerk!

Floppy 1581: ideales Zweitlaufwerk

Fazit: Es ist nicht alles Gold, was glänzt - vor allem auf der Test/Demo-Diskette zur 1581. Bis auf die genannten Blindgänger entsprechen die Tools und Utilities den Anforderungen der 1581. Dennoch sollte sich ein geübter Programmierer nochmals mit den Programmen und Files beschäftigen und sie überarbeiten, auch wenn's nur eine Gratisdiskette ist. Für viele Käufer der Floppy 1581 ist es nämlich die erste Connection zu dieser fantastischen Diskettenstation.

Sieht man sich auf dem Software-Markt um, muß man schon mit der Lupe suchen, um kommerzielle Software auf einer 3½-Zoll-Diskette zu entdecken (unsere Sonderheftdisketten nicht ausgenommen): Daher möchten wir Ihnen abraten, eine 1581 als einziges Laufwerk zum C 64 anzuschaffen. Damit können Sie lediglich eigene Programmentwicklungen oder Daten speichern und laden. Gute Profi-Software oder Spiele bleiben Ihnen (bis dato) versagt: Die Hersteller benutzen zu 99,9 Prozent 5¼-Zoll-Scheiben. Geradezu perfekt aber präsentiert sich die 1581 in ihrer Funktion als unterstützendes Zweitlaufwerk (z.B. Nr. 9) - vor allem mit Geos V2.0. Alle Diskettenverwaltungs-Funktionen des Desktop (das sich aber selbstverständlich nur von der 5¼-Zoll-Disk in einer 1541-Floppy mit Geräteadresse 8 booten läßt!). Der technische Standard überbietet die 1541 oder 1571, was sich z.B. in der schnelleren Verarbeitungsgeschwindigkeit durch den Caché-Speicher ausdrückt.

Floppy 1581 (Technische Daten)

Hardware:	Typbezeichnung Mikrochip
Mikroprozessor:	6502A
I/0-Chip:	CIA 8520A
32 KByte ROM:	23256
8 KByte RAM:	4364
Disk-Controller:	WD 1772
Speicherkonfiguration	
Kapazität unformatiert:	1 MByte (1 024 000 Byte)
Kapazität formatiert:	790 KByte (808 960 Byte)
maximale Datensätze pro Datei:	65 535
maximale Einträge ins Directory:	296
Zylinder pro Diskette:	80
logische Sektoren pro Zylinder:	40 (0 bis 39)
physikalische Sektoren pro Zylinder:	20 (1 bis 20)
Sektoren pro Diskette:	3160
logische Bytes pro Sektor:	256
physikalische Bytes pro Sektor:	512
(Auszug Anhang E des Bedienungshandbuchs zur 1581, Art.-Nr. 581581, 1986)	

Und wo gibt's das Laufwerk-Wunder? Hier zwei Adressen, die Sie sich notieren sollten:

plus-electronic GmbH, Marienstr. 2, 3016 Seelze 1, Tel. (05137) 5 04 77, Best. Nr. 280 140-9, 375 Mark (Vorzugspreis für Altkunden: 359,50 Mark - nur solange Vorrat reicht!),

Scanntronik GmbH, Parkstr. 38, 8011 Zorneding, Tel. (08106) 2 25 70, 358 Mark (+ 8 Mark Versandkosten).

Letzte Meldung: Beim Geos User Club (Heinisch & Haberland) ist die Floppy 1581 bereits ausverkauft! Wie uns die anderen Vertriebsfirmen telefonisch mitteilten, sind dort Geräte in ausreichender Menge vorhanden. (bl)

Dual-Filecopy – 5¼-Zoll-Disketten zur 1581 transferieren

Datenschaufler

Die meisten Kopierprogramme haben ein Manko: Sie arbeiten nur mit einem Laufwerk! Abgesehen von lästigen Diskettenwechseln: haben Sie schon einmal versucht, eine 3½-Zoll-Diskette in die 1541 oder 1571 zu schieben?

Wer bisher Daten von 5¼-Zoll-Scheiben auf die kleineren Micro-Disketten des Zweitlaufwerks 1581 übertragen wollte, stand im Regen: Da blieb nur LOAD und SAVE. War's bei Basic-Programmen zumindest zeitraubend, wußten viele bei Assembler- oder Hires-Grafikdateien nicht mehr weiter.



Dual-Filecopy: Bequeme Optionswahl per Funktionstasten

Endlich ist das Problem gelöst: »Dual-Filecopy« arbeitet mit **zwei** Laufwerken (Quellfloppy 8 und Ziel-Diskettenstation 9), die unterschiedlichen Typs sein dürfen: z.B. von der 1541 zur 1571, zwischen zwei 1541-Geräten oder zwei 1571ern usw.

1581 muß Ziel-Floppy sein!

Und das wichtigste: Die heikle 1581 gebärdet sich ebenfalls nicht aufmüpfig und akzeptiert alle Programmfunktionen! Lediglich eine Vorschrift sollte man beachten: Die 3½-Zoll-Floppy 1581 muß unbedingt auf Geräteadresse 9 eingestellt sein (per DIP-Schalter an der Gehäuserückseite).

Laden Sie das Kopier-Tool mit:

```
LOAD "DUAL-FILECOPY",8
```

und starten Sie es mit RUN.

Das Zweitlaufwerk muß bei Programmstart bereits angeschlossen und aktiv sein, sonst erhält man die Meldung »Hardwarefehler!«. Der Menübildschirm erscheint (Abb.). Sämtliche Optionen ruft man mit den entsprechenden Funktionstasten auf:

<F1> **Copy Files #8 to #9**: ... leitet die Kopieraktion ein (von Laufwerk 8 zu 9). Bedenken Sie: »Dual-Filecopy« ist kein Backup-Programm und besitzt keine automatische Formatierfunktion! Bei blanken, neuen Disketten oder Disketten, die Sie formatieren möchten, muß man vorher Menüpunkt <F5> aktivieren (s. Beschreibung).

Schieben Sie nun eine Quell- und Zieldiskette (mit ausreichend freier Blockzahl!) in die Laufwerke und drücken Sie eine Taste. Das Programm holt das Directory und fragt bei jeder Datei, ob sie kopiert werden soll (j/n), <RETURN> beendet die Abfrage. Dazu ein Hinweis: Trennstriche des Directory (die gibt's z.B. auf allen Disketten der 64'er-Sonderhefte), mit 0 Blocks und als USR- bzw. DEL-Filetyp gekennzeichnet, sollten Sie aus Gründen der speziellen Datenablage niemals auswählen (also <N> drücken!), damit die BAM (Blockbelegungsplan) beim Kopieren korrekt auf Disektte geschrieben wird. <J> bei der Frage »ok?« startet die Kopieraktion, <N> wiederholt die Auswahl. Auf Wunsch kann man den Bildschirm ausblenden (Screen blanking).

»Dual-Filecopy« akzeptiert nur die File-Typen PRG, SEQ und USR. Andere Dateien (z.B. REL) kann das Programm nicht korrekt kopieren und überspringt diese mit einer entsprechenden Meldung.

Kurzinfo: Dual-Filecopy

Programmart: Kopierprogramm für Einzeldateien mit zwei Laufwerken
Laden: LOAD "DUAL-FILECOPY",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: funktioniert mit den Commodore-Floppies 1541, 1541-II, 1570, 1571 (auch internes Laufwerk des C128D) und 1581, auch verschiedene Laufwerkstypen untereinander.
Benötigte Blocks: 12
Programmautor: Thomas Hohenberger

Erscheint der Hinweis »Copy complete«, kommt man per Tastendruck wieder ins Menü.

<F2> **Scratch Files #8**: ... löscht beliebig viele Dateien auf einer Diskette im Quellaufwerk 8. Die Dateiabfrage ist identisch mit der von Option <F1> (Kopieren), anschließend werden alle ausgewählten Files in einem Rutsch entfernt.

DOS 10.0-Befehle werden akzeptiert

<F3> **Disk Error #8**: ... liest den Fehlerkanal des Quelllaufwerks (Nr. 8),

<F4> **Disk Error #9**: ... zeigt, welcher Floppyfehler sich bei Laufwerk 9 eingeschlichen hat,

<F5> **Order #8**: ... akzeptiert DOS-Befehle für die Floppy Nr. 8: Alle üblichen Anweisungen wie N, V, I, S, C und R werden in der korrekten Schreibweise lt. Handbuch angenommen, ebenso der </>-Befehl (Partition) für die 1581 (um Dateien z.B. in ein 1581-Subdirectory zu speichern). Man spart sich die OPEN- und CLOSE-Anweisungen.

<F6> **Order #9**: ... wie Menüpunkt <F5>, aber fürs Ziellaufwerk. Ein Beispiel zum Formatieren der Zieldisk in Floppy 9 (mit ID-Kennung):

```
Order Drive #9: n:copydisk,cp
```

<F7> **Directory #8**: ... zeigt das Inhaltsverzeichnis der Diskette im Quellaufwerk,

Kopieren ohne Diskettenwechsel

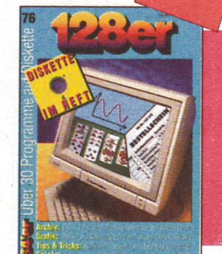
<F8> **Directory #9**: dito, fürs Ziellaufwerk.

»Dual-Filecopy« gehört sicher nicht zu den schnellsten Kopierprogrammen, arbeitet aber problemlos mit der 1581 zusammen. Außerdem entfällt der lästige Wechsel zwischen Quell- und Zieldisketten: Bei Einzellaufwerk-Kopierprogrammen kann das nämlich bei großer Datenmenge fast in Arbeit ausarten! Last not least: »Dual-Filecopy« schlägt das auf der Commodore-Startdisk zur 1581 mitgelieferte »Unicopy« um etliche Längen (ganz zu schweigen von dessen miserabler Bildschirmausgabe!). (bl)

64'er Sonderhefte alle auf einen Blick

Die 64'er Sonderhefte bieten Ihnen umfassende Information in komprimierter Form zu speziellen Themen rund um die Commodore C 64 und C 128. Ausgaben, die eine Diskette enthalten, sind mit einem Diskettensymbol gekennzeichnet.

C 64, C 128, EINSTEIGER



SH 76: C 128
"DiskEtti 128" druckt Diskettenaufkleber/ Mehr Sprites mit "Sprite-Tool"



SH 22: C128 III
Farbiges Scrolling im 80-Zeichenmodus
8 Sekunden-Kopierprogramm



SH 26: Rund um den C64
Der C64 verständlich für Alle, mit ausführlichen Kursen



SH 36: C 128
Power 128: Directory komfortabel organisieren / Haushaltsbuch: Finanzen im Griff / 3D-Landschaften auf dem Computer



SH 38: Einsteiger
Alles für den leichten Einstieg / Super Malprogramm / Tolles Spiel zum selbermachen / Mehr Spaß am Lernen



SH 50: Starthilfe
Alles für den leichten Einstieg / Heiße Rythmen mit dem C 64 / Fantastisches Malprogramm



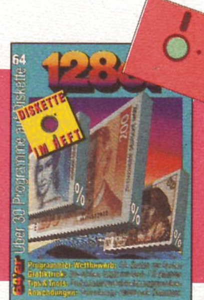
SH 51: C 128
Volle Floppy-Power mit "Rubikon" / Aktienverwaltung mit "Börse 128"



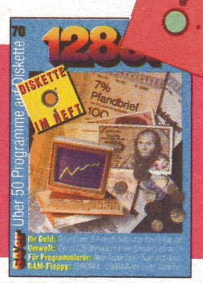
SH 58: 128er
Übersichtliche Buchhaltung zuhause / Professionelle Diagramme



SH 62: Erste Schritte
RAM-Exos: Disketten superschnell geladen/ Exbasic Level II: über 70 neue Befehle/ Raffinessen mit der Tastatur



SH 64: 128er
Anwendungen: USA Journal / Grundlagen: CP/M, das dritte Betriebssystem/ VDC-Grafik: Vorhang auf für hohe Auflösung



SH 70: C 128
Finanzen/ Vereinsverwaltung/ Umwelt/ CP/M-Grundlagen/ Hardware/ Tips&Tricks



SH 74: Einsteiger
Basic 3.5: über 40 neue Befehle und Tastaturfunktionen/ FOBS: Komfortable Benutzeroberfläche/ Tips&Tricks/ Open Access: Dateiverwaltung, Videos, Adressen usw.

PROGRAMMIERSPRACHEN



SH 35: Assembler
Abgeschlossene Kurse für Anfänger und Fortgeschrittene



SH 40: Basic
Basic Schritt für Schritt / Keine Chance für Fehler / Profi-Tools und viele Tips



SH 71: Assembler
Kursus/ Komplettpaket/ Befehlsposter/ Tips&Tricks/ Leserfragen

DTP



SH 39: DTP, Textverarbeitung
Komplettes DTP-Paket zum Abtippen / Super Textsystem / Hochauflösendes Zeichenprogramm

ANWENDUNGEN



SH 78: Anwendungen
Business-Grafik: Statistik zum Anfassen / Raffinierter Soundeditor und 15 Demos / Mit MAS 1.0 zum Einser Abitur

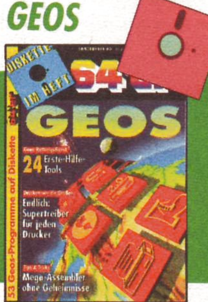


SH 81: Anwendungen
Zeichenprogramm der Superlative: Paint-Mania 64 / Disketten im Griff: Disk-Tool V 6.5 / Der Knopfdruck-Komponist: Maestro 64



SH 68: Anwendungen
Kreuzworträtsel selbstgemacht/ Happy Synth: Super-Synthesizer/ Sir-Compact: Bit-Packer verdichtet Basic- und Assemblerprogramme.

GEOS

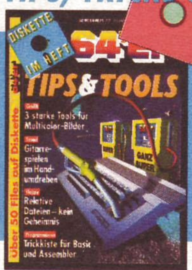


SH 80: GEOS
24 Erste-Hilfe-Tools / Supertreiber für jeden Drucker / Mega-Assembler ohne Geheimnisse

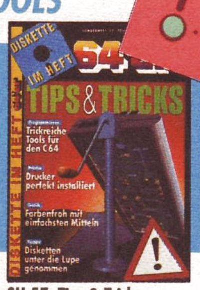


SH 59: GEOS
GeoBasic: Großer Programmierkurs mit vielen Tips & Tricks

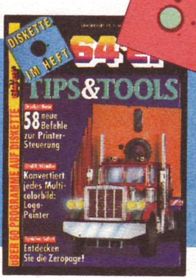
TIPS, TRICKS & TOOLS



SH 77: Tips&Tools
Grafik: Tools für Multicolor-Bilder / Tricks für Basic und Assembler / Floppy: Relative Dateien - kein Geheimnis



SH 57: Tips & Tricks
Trickreiche Tools für den C64 / Drucker perfekt installiert



SH 65: Tips&Tools
Streifzug dirch die Zeropage/ Drucker-Basic: 58 neue Befehle zur Printer-Steuerung/ Multicolorgrafiken konvertieren/ über 60 heiße Tips&Tricks

HARDWARE



SH 25: Floppylaufwerke
Wertvolle Tips und Informationen für Einsteiger und Fortgeschrittene



SH 47: Drucker, Tools
Hardcopies ohne Geheimnisse / Farbige Grafiken auf s/w-Druckern



SH 67: Wetterstation:
Temperatur, Luftdruck und feuchte messen/ DCF-Funkuhr und Echtzeituhr/ Daten konvertieren: vom C64 zum Amiga, Atari ST und PC

GRAFIK



SH 75: Grafik
Superfrac: Welt der Fractale / Hi-Eddi: Zeichenprogramm der Spitzenklasse



SH 45: Grafik
Listings mit Pfiff / Alles über Grafik-Programmierung / Erweiterungen für Amica-Paint

SH 63: Grafik
Text und Grafik mischen ohne Flimmern / EGA: Zeichenprogramm der Superlative / 3 professionelle Editoren

SPIELE





Top Spiele 1
Die 111 besten Spiele im Test/ Tips, Tricks und Kniffe zu heißen Games/ Komplettlösung zu "Last Ninja II"/ große Marktübersicht: die aktuellen Superspiele für den C64



SH 30: Spiele für C64 und C128
Spiele zum Abtippen für C64/ C128/ Spieleprogrammierung



SH 37: Spiele
Adventure, Action, Geschicklichkeit / Profihilfen für Spiele / Überblick, Tips zum Spielekauf



SH 42: Spiele
Profispiele selbst gemacht / Adventure, Action, Strategie



SH 49: Spiele
Action, Adventure, Strategie / Sprites selbst erstellen / Virenkiller gegen verseuchte Disketten



SH 52: Abenteuerspiele
Selbstprogrammieren: Von der Idee zum fertigen Spiel / So knacken Sie Adventures



SH 54: Spiele
15 tolle Spiele auf Diskette/ der Sieger unseres Programmierwettbewerbs: Crillion II/ ein Cracker packt aus: ewige Leben bei kommerziellen Spielen

SH 60: Adventures
8 Reisen ins Land der Fantasie - so macht Spannung Spaß



SH 61: Spiele
20 heiße Super Games für Joystick-Akrobaten/ Cheat-Modi und Trainer POKEs zu über 20 Profi-Spielen/ Krieg der Kerne: Grundlagen Spieleprogrammierung

SH 66: Spiele
15 Top-Spiele mit Action und Strategie/ Mondlandung: verblüffend echte Simulation und Super-Grafik/ High-Score-Knacker: Tips&Tricks zu Action-Games



SH 73: Spiele
Action bis Adventure: Zehn Spiele zum Kampf gegen Fabelwesen/ Preview/ Tips&Tricks/ Kurse/ Game Basic/ Mission II/ W.P. Tennis II/ Omnibus GmbH/ Mic's Push'em



SH 79: Spiele
25 superstarke Spiele. Action, Geschicklichkeit, Strategie und die Mini-Parade. Mit diesen Tips&Tricks knacken Sie jedes Spiel.

64'er Magazin auf einen Blick

Diese 64'er-Ausgaben bekommen Sie noch bei Markt&Technik für jeweils 7,-DM. Ab Ausgabe 1/92 kostet das Heft 7,80 DM. Die Preise für Sonderhefte und Sammelbox entnehmen Sie bitte dem Bestellcoupon. Tragen Sie Ihre Bestellung im Coupon ein und schicken Sie ihn am besten gleich los, oder rufen Sie einfach unter 089 - 240 132 22 an.

7/91: Trickfilm mit dem C 64 / Bauanleitung: 1541-Floppy mit Batteriebetrieb / Listing des Monats: Basic-Butler

8/91: Drucker unter 1000 DM / Test: GEO-RAM / Listing des Monats: 80-Farben-Malprogramm / Longplay: Secret of the Silver Plate

9/91: Joystick im Test / Die üblen Tricks mit Raubkopien / Die besten Drucker unter 1500 DM / Mit großem Spieleteil

10/91: 100 besten Tips&Tricks / Listing: Fraktal-Programm / C-64-Meßlabor: komfortables Kontrollmodul

11/91: Alles über Diskette & Floppy / Bauanleitung: C 64 steuert Laserstrahl / Sha-Jongg: Topspiel mit Spitzengrafik / Großer Spieleteil

12/91: Alle Spiele 1991 / Tolle Tips&Tricks für den C64 und C128 / Geschenktips für Computerfans / Komfortable Videoverwaltung

1/92: Viren/ Die neue 64er Floppy/ Neue Produkte-Top-Tests/ Floppy-Kursus für Fortgeschrittene/ Assembler-Corner

5/92: Desktop-Publishing: Alles über DTP, Test DTP-Programme / Scanner: So holt man Bilder in den Computer / Programm des Monats: Top-Adreßverwaltung

6/92: Software auf Knopfdruck: Alles über EPROMs / Datenkonvertierung vom C64 zu Amiga, PC & Atari ST / Programm des Monats: Magazin-Creator de Luxe

7/92: 64er Jubiläum: Von '82 bis '92 / Knallharte Tests: Flüster-Drukker, Geos-Software etc. / Top-Listing: Line V1.0 - Grafik vom Feinsten

8/92: Test: 8 Top-Drucker unter 600 Mark / Hardware: C64 an 12 Volt-Batterie / Daten und Adreßanzeige selbst gebaut / Jede Menge Programme und Tips&Tricks

9/92: Die Besten Joysticks: Newcomer aus England und großer Vergleichstest / Drucker unter 1000 DM auf dem Prüfstand / Assembler für Einsteiger /35 Seiten Tips & Tricks

10/92: Perfekte Filme mit dem C64 / Alle Zeichen- und Malprogramme / Die Kopierschutztricks der Profis / Tests: Drucker-Interfaces, Joystick-Stars (II)

BESTELLCOUPON

Ich bestelle 64er Sonderhefte

___ Sonderhefte ohne Diskette je 14,- DM ___/___/___/ ___ DM
___ Sonderhefte mit Diskette je 16,- DM ___/___/___/ ___ DM
___ Sonderhefte 51, 58 und 64 je 24,- DM ___/___/___/ ___ DM
___ Sonderheft "Top Spiele 1" 9,80 DM DM

Ich bestelle ___ 64er Magazin Nr. ___/___/___/ ___ DM
je Heft 7,- DM, ab Ausgabe 1/92 je Heft 7,80 DM

Ich bestelle ___ Sammelbox(en) ___ DM
zum Preis von je 14,- DM

Gesamtbetrag ___ DM

Ich bezahle den Gesamtbetrag zzgl. Versandkosten nach Erhalt der Rechnung.

Name, Vorname

Straße, Hausnummer

PLZ, Wohnort

Schicken Sie bitte den ausgefüllten Bestellcoupon an: 64er Leserservice, CSJ, Postfach 140 220, 8000 München 5, Telefon 089/ 240 132 22

Lokaltermin

Im Kopierparadies

Was geschieht mit der Originaldiskette zum aktuellen 64'er-Sonderheft, wenn sie die Redaktion verlassen hat? Wir haben uns an ihre Hülle geheftet...

Wer bisher geglaubt hat, wir würden die Disketten zu den 64'er-Sonderheften in Tag- und Nachtschichten per Backup-Programm (z.B. Track-Copy II) und einem 1541-Laufwerk selbst in den Redaktionsräumen vervielfältigen, liegt völlig schief: Die Redaktion stellt lediglich die entsprechenden Dateien auf der Disk zusammen, prüft, ob sie funktionieren und durchsucht die Originaldisk (Master-Diskette) nach physikalisch fehlerhaften Spuren oder Sektoren (Error Scan).

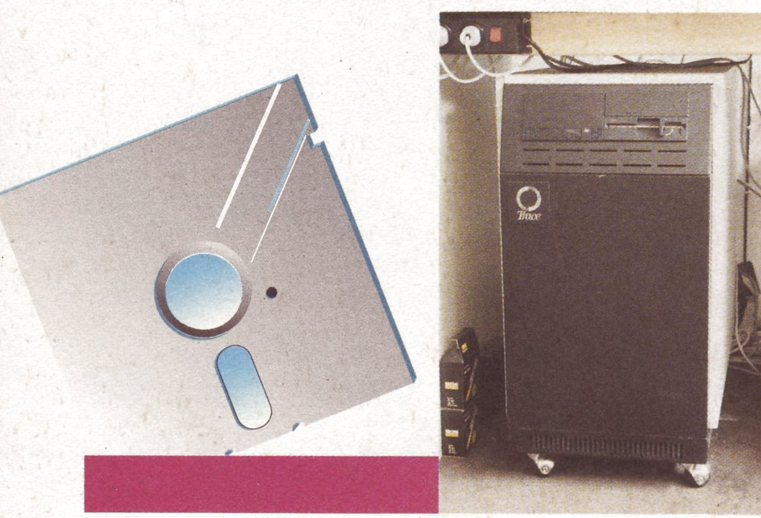
War alles in Ordnung, ist sie ab sofort dem redaktionellen Einfluß entzogen: Sie landet termingerecht in den Arbeitsräumen des Duplikations-Service München (kurz: DUP-Center).

Dort reiht sich die 64'er-Master-Disk in die lange Schlange anderer Originaldisketten ein, die darauf warten, zigtausendmal vervielfältigt zu werden: Pro Monat verarbeitet das DUP-Center etwa 300 000 Disketten (vorwiegend 3½-Zoll-Scheiben für PCs/ATs, Amiga und Apple Macintosh, aber auch 5¼-Zoll-Disketten der 64'er-Redaktion und für PCs).

Diese fantastischen Zahlen erreicht man nur mit einer technischen Ausrüstung, die »top« ist: Der Steuermann (fast) aller Arbeitsabläufe ist der »Trace Mountain«, ein Computer, bestückt mit dem Betriebssystem Unix (Abb. 1). Der Rechner läuft mit dem Unix-fähigen Motorola-Mikroprozessor 68030 (den hat auch der Amiga 3000), wurde in den USA entwickelt, besitzt 8 MByte RAM-Speicher (aufrüstbar) und eine 60-MByte-Festplatte. Ausgerüstet ist er mit einem Dual- und High-Density-Floppy-Controller und kostet »nackt« ca. 50000 Mark. Der DFC (Dual Floppy Controller) schafft z.B. 240 PC-Disketten à 360 KByte Speicherkapazität in der Stunde, der HDFC (High Density) immerhin 120 Scheiben mit 1,2 (5¼-Zoll) bzw. 1,44 MByte (3½-Zoll).

Er läßt sich intern mit maximal zehn Zusatzplatinen (Floppycontrollern) bestücken, die beim Kopieren zehn verschiedene Diskettenlader steuern (Abb. 2). Durch die Controller erhöhen sich die Kosten der Kopieranlage dramatisch: etwa 200000 Mark zusätzlich! Aufgrund des relativ geringen Arbeitsspeichers (8 MByte RAM) sollte man aber höchstens drei unterschiedliche Diskettenformate mit verschiedenen Dateninhalten gleichzeitig duplizieren. Es ist nämlich ohne weiteres möglich, gleichzeitig eine PC/AT-, Amiga- und Atari-Master-Disk auf verschiedenen, dafür eingeteilten Ladergeräten zu kopieren! Unter den DUP-Centern Europas gilt der Trace und dessen Technologie als Marktführer.

Der Unix-Computer besitzt eine speziell entwickelte Steuersoftware (»Series 1000 Diskettes Duplicator«), die das Einlesen der Master-Disk-Daten und die Übertragung auf Leerdisketten im Lader übernimmt: 50 verschiedene Diskettenformate stehen zur Verfügung, das passende wird eingestellt.



[1] Unix-Computer mit Spezialauftrag: Trace-Mountain mit 68030-Mikroprozessor

50 verschiedene Diskettenformate einstellbar

Jetzt kommt die Master-Disk in den Lader, deren Daten auf beiden Diskseiten in **einer** Datei ins RAM des Trace übertragen werden (der Computer markiert automatisch, wo die erste Seite aufhört und die Rückseite beginnt!). Die 340 KByte unserer zweiseitigen Sonderheftdiskette haben sich inzwischen durch die Verarbeitung des Unix-Programms zu einem 4 MByte großen Daten-File entwickelt. Da zum Kopieren aber noch andere Vorbereitungen zu treffen sind, wird diese Datei (z.B. »SH 83«) zunächst zwischengespeichert: Dazu dient ein IBM-kompatibler MS-DOS-Computer mit 120-MByte-Festplatte, der mit dem Unix-Trace vernetzt wurde. Um Speicherplatz zu sparen, wird die 4 MByte große Datei der Sonderheftdiskette im Unix-Computer während des Übertragungsvorgangs gepackt – zum Schluß bleibt noch ein File mit etwa 65 KByte übrig, das auf der Festplatte des IBM-kompatiblen AT abgelegt wird.

Wenn nach Terminplan unsere Sonderheftdiskette an der Reihe ist, sucht der Operator die komprimierte Datei auf der MS-DOS-Festplatte und schickt sie zum Trace zurück: Jetzt wird das File bei der Übertragung entpackt und landet wieder als 4-MByte-Datei im RAM des Unix-Computers. Theoretisch könnte es nun mit der Kopieraktion losgehen, wenn man zuvor die Anzahl der einzusetzenden Lader (z.B. drei) bestimmt hat, aber: Die Hardware-Voraussetzungen müssen ebenfalls stimmen! Da es z.B. sein kann, daß die Lader-Geräte vorher 5¼-Zoll-Disketten mit 1,2 MByte für IBM-ATs kopiert haben, hätte kein Byte unserer Diskette im Commodore-GCR-Format



[2] In Reih und Glied: Disketten stapeln sich in den Laderschächten.

Auswechselbare Laufwerke für die Lader

überhaupt den Hauch einer Chance, korrekt auf eine zweite Disk übertragen zu werden. Das Problem ist rasch gelöst: Das Lader-Laufwerk (sieht übrigens den internen PC-Floppies täuschend ähnlich) wird mit wenigen Handgriffen ausgetauscht!

Bei den auswechselbaren Lader-Floppystationen zu unseren Sonderheftdisketten handelt es sich um technische Kostbarkeiten (abgesehen vom Preis: ca. 3000 Mark pro Stück): Sie kopieren beide Diskettenseiten gleichzeitig (Abb. 3)!

Viele werden jetzt abwinken und entgegnen, das sei doch nichts Besonderes: Das macht ja schon die C-128-Floppy 1571 im Normalzustand! Irrtum - die 1571 bearbeitet im 128er Modus zwar beide Seiten einer $5\frac{1}{4}$-Zoll-Diskette (formatieren, speichern, kopieren usw.), aber das Ergebnis ist eine **doppelseitige** Disk mit nur einer Directory-Spur (Track 18 auf der Vorderseite). Diesem Prinzip folgen z.B. auch PC/AT-Laufwerke oder die 1581 mit ihren $3\frac{1}{2}$-Zoll-Disketten.

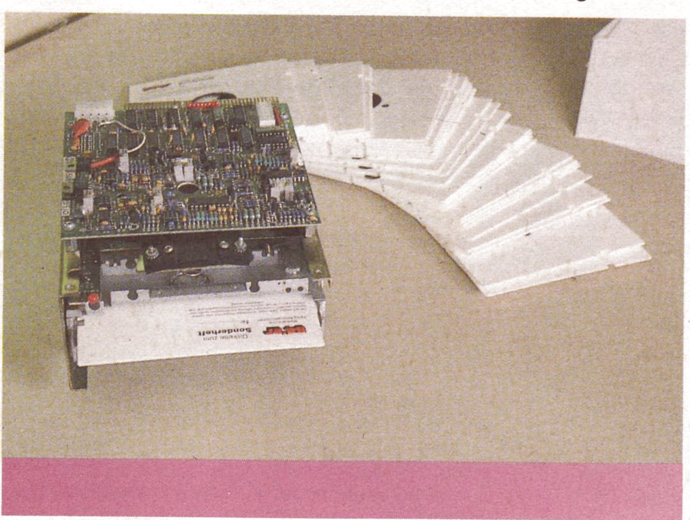
Unsere Sonderheftdisk aber ist **zweimal einseitig (zweiseitig)** bespielt, also mit zwei in sich geschlossenen Diskettenseiten und jeweils eigenem Directory (als wären's zwei separate Disketten!). Der Unterschied:

- doppelseitige Disketten: ...lassen sich nur mit Laufwerken korrekt lesen und beschreiben, die zwei Schreib-Leseköpfe besitzen. Geht die Speicherkapazität auf Seite 1 zur Neige

Die zweite Diskseite wird spiegelverkehrt kopiert!

(dort ist Kopf 1 zuständig), wird floppyintern automatisch Kopf 2 eingesetzt, der ohne Unterbrechung und an der richtigen Stelle auf der Diskseite 2 weitermacht. Bei doppelseitigen Disketten kommt's z.B. oft vor, daß ein logisch zusammenhängendes Programm physikalisch teilweise auf der Vorder-, der Rest aber auf der Diskrückseite gespeichert ist: Beim Laden dieser Datei (mit der 1571 oder einem PC-Laufwerk) merken Sie überhaupt nichts von dieser Trennung!

- zweiseitige Disketten: ...sind lediglich zwei einseitige Disketten, auf Vorder- und Rückseite einer Scheibe zusammengefaßt. Zum Lesen und Schreiben läßt sich nur der Schreib-Lesekopf für Seite 1 einsetzen.



[3] Das Speziallaufwerk kopiert Flippies (zweiseitige $5\frac{1}{4}$-Zoll-Disketten)

In der Fachsprache der DUP-Center nennt man diesen Diskettentyp »Flippy« (flip = umdrehen). Vor Jahren war's noch umständlich mit den Flippies: Disketten-Duplizierbetriebe mußten erst die Vorderseite kopieren, die Daten der Rückseite einlesen, dann die Disketten umdrehen, erneut in den Schacht des Laders legen und die Daten auf die Rückseite übertragen. Die Wunderlaufwerke unseres DUP-Centers machen's anders: Sie besitzen ebenfalls zwei Schreibköpfe. Der zweite wurde aber speziell vorbereitet und eingestellt, um die Daten - später korrekt lesbar (fürs Laufwerk des Laders also spiegel- und spurverkehrt!) - übertragen zu können: Die Diskette wird ja nicht gewendet! Nur so ist es möglich, beide separaten Seiten einer zweimal einseitigen Disk 1 : 1 zu übertragen.

Derzeit gibt's in Europa nur einen Techniker, der diesen Trick beherrscht - er steht in Diensten des Herstellers der Trace-Moutain-Geräte...

Allerdings zahlt sich die Anschaffung solcher Speziallaufwerke für DUP-Center aus: Pro Stunde und Lader lassen sich so bis zu 120 korrekt kopierte Flippies produzieren.

Der Einfüllschacht eines Laders faßt 75 $5\frac{1}{4}$-Zoll-Disketten pro Durchgang. Die Schächte werden beim Kopieren ständig beobachtet und rechtzeitig nachgefüllt. Nach dem Kopiervorgang kommen die fertigen Duplikate wieder in den Lieferkarton des Diskettenherstellers (unformatiert und ohne Hülle werden sie 500-stückweise angeliefert) und wandern zum nächsten Arbeitsgang: Labeln (Etikettaufkleber anbringen).

Disketten automatisch umhüllen

Per Hand kleben ist schon längst out - das übernimmt eine schnelle Etikettiermaschine: Disketten oben reinlegen, Endlospapier mit den Disklabels einspannen und starten - unten kommen die Scheiben mit exakt plaziertem Diskaufkleber raus! Das Funktionsprinzip wurde den Briefadressen-Etikettiermaschinen entliehen: Außer technischen Veränderungen des Einzugschachts (Disketten sind dicker als Briefe) mußte praktisch nichts geändert werden.

Die Scheiben sind jetzt fast verkaufsfertig - es fehlt nur noch die Hülle. Die Kartons bringt man nach dem Labeln zu einer Vorrichtung mit langem Fließband: Dort erhalten die wertvollen schwarzen Scheiben ihren Schutzumschlag.

Endlich können die Disketten verpackt und die Kartons zugeklebt werden: Jetzt geht's zur Druckerei, in der die Disketten dann während des Bindevorgangs maschinell auf Seite 19 des aktuellen Sonderhefts geklebt werden. Nach dem Verpacken (25 Stück pro Paket) stehen die 64'er-Sonderhefte zur Abholung für den Zeitungsgrossisten bereit.



[4] Das DUP-Center-Team: Mehr als 300 000 Disketten werden pro Monat verarbeitet.

Der wiederum beliefert Kioske und den Zeitschriftenhandel nach einem ausgeklügelten Vertriebssystem.

Ein Menge Arbeit, die das DUP-Center-Service-Team (Abb. 4) hinter sich hat, bis Sie Ihre Sonderheftdiskette in die eigene Floppystation schieben können, stimmt's? (bl)

Wir danken Hans-Peter März vom Duplication- und Service-Center, Hans-Pinsel-Str. 2, 8013 Haar, Tel. 089/46 13-0, für die freundliche Unterstützung.

Fachbegriffe zu Disketten und Laufwerken

Kleines Floppy-lexikon

Manche behaupten, Computer-Chinesisch sei schwerer zu erlernen als Kisuaheli. Sei's drum: Unsere Übersicht erklärt verständlich oft verwendete Schlagwörter, die beim Umgang mit der Floppy und Massenspeichern immer wieder auftauchen.

Block Availability Map (BAM): (Blockbelegungsplan in Spur 18, Sektor 0). Daraus ist ersichtlich, welche Sektoren noch frei oder benutzt sind. Außerdem informiert die BAM über Namen, ID, das Formatkennzeichen der Disk und über den Beginn des ersten Directory-Sektors (s. Tabelle 1).

Block: (auch Sektor)...ist ein 256 Byte großer Bereich innerhalb einer Diskettenspur, der zur Datenspeicherung verwendet wird. Nur 254 Byte sind frei: Die beiden ersten Speicherstellen eines Blocks enthalten Spur- und Sektornummer des logisch folgenden Blocks der aktuellen Datei (z.B. ein Basic-Programm). Automatische Verwaltung durchs Floppy-DOS.

Burst-Modus: erweiterter Floppy-Befehlssatz zur schnelleren Datenmanipulation. Funktioniert nur mit den Disk-Controllern der 1570/1571 (WD 1770) und 1581 (WD 1772).

Directory: Inhaltsverzeichnis der auf der Disk gespeicherten Dateien. Beginnt bei den Floppies 1541/1571 und 5¼-Zoll-Disketten in Spur 18, Sektor 1 (oder lt. Blockbelegungsplan), bei einer 3½-Zoll-Disk der 1581 liegt das Verzeichnis in Spur 40. Die 1541/1571-Floppies können maximal 144 Namen ins Directory eintragen, die 1581 dagegen 296. Tabelle 2 entschlüsselt die Bedeutung der Bytes im Sektor.

Byte-Belegung der BAM (Spur 18, Sektor 0)

Byte-Position:	Funktion
0:	Spur des ersten Directory-Sektors (normal: 18)
1:	Nummer des ersten Directory-Sektors (normal: 1)
2:	Formatkennzeichen (bei 1541/1570/1571 immer $41 = A). Jede andere Kennung (.z.B. B) versiegelt die Diskette (Schreibschutz).
3:	Anzahl der Diskettenseiten ($00 = einseitig, z.B. 1541; $80 =doppelseitig, z.B. 1571).
4:	freie Sektoren in Spur 1 (z.B. $15 = alle 21 Datenblöcke frei).
5:	Bitmuster der Blockbelegung (Sektoren 0 bis 7 in Spur 1). $FF (255) = alle Sektoren frei.
6:	Sektoren 8 bis 16.
7:	Sektoren 17 bis 20
8 bis 11:	entspricht den Funktionen der Bytes 4 bis 7, aber für Spur 2.
...	
140 bis 143:	...für Spur 35.
144 bis 161:	Diskname, aufgefüllt mit CHR$(160) = <SHIFT SPACE>. Obwohl es 18 Byte sind, darf der Diskname nie länger als 16 Zeichen sein!
162/163:	ID-Kennung (2 Byte).
164:	CHR$(160) <SHIFT SPACE>.
165/166:	Formatangabe zur Diskette (bei 1541 und 1571 = 2A).
167 bis 170:	CHR$(160)
171 bis 179:	$00-Bytes bei der 1541, $A0 mit den 1570/1571-Floppies.
180 bis 220:	nicht benutzt,
221 bis 255:	nur bei doppelseitig formatierten Disketten der 1571: Anzahl der freien Blöcke der Spuren 36 bis 70. Bei der 1541 hat dieser Bereich keine Bedeutung.

Tabelle 1. Informationen für die Floppy: die BAM

Double Sided (DS, 2S): Diskette ist doppelseitig bespielbar. Die Bezeichnung ist irreführend, denn: Jede im Handel befindliche Diskette ist auf beiden Seiten magnetisch beschichtet, kann also nach dem Formatieren beidseitig mit Daten beschickt werden. Nur: Der Hersteller vergibt mit dem Qualitätsprädikat DS (2S) quasi eine Garantie, daß es sich auf beiden Seiten problemlos speichern läßt – im Gegensatz zu SS (1S): Hier übernimmt er nur die Gewährleistung für die Diskvorderseite.

DOS: (Disk Operating System), Betriebssystem des Diskettenlaufwerks, das Befehle des Anwenders entgegennimmt und ausführt. Bei den Floppies 1541/70/71/81 umfaßt es 32 KByte und beginnt an der internen Floppyspeicheradresse $8000 (32768). Da sich diese Systemsoftware in einem separaten Mikrochip befindet, kann man die Daten auslesen, ändern und das neue DOS in ein EPROM brennen. Anschließend muß man die Chips im Gerät austauschen.

Direktzugriffsdatei: Daten werden auf beliebig freie Sektoren einer formatierten Diskette gespeichert (DOS-Befehl Block-Write [B-W], bzw. U2). Dazu müssen Spur- und Sektornummer angegeben werden, ebenso beim Lesen per DOS-Anweisung Block-Read (B-R) bzw. U1. Der Vorteil: Da kein Eintrag im Directory erscheint, sind solche Dateien auf Disk nicht sichtbar – also nie vergessen, in welchen Spuren und Sektoren man suchen soll! Es empfiehlt sich, nach Anlegen einer Direktzugriffsdatei die verwendeten Sektoren in der BAM als belegt zu kennzeichnen (mit Block-Allocate [B-A]).

Disk-Controller (DC): Hardware (Mikrochip) in der Floppystation, der für den korrekten Laufwerksbetrieb zuständig ist. Die Floppy 1571 besitzt einen PC-Laufwerk-kompatiblen DC (Typ WD 1770), ebenso die 1581 (Typ WD 1772). Die 1541 verwendet ein Logic-Array (simpler Custom-Chip) als Disk-Controller.

Aufbau eines Directory-Sektors (ab Spur 18, Sektor 1)

Byte-Position:	Bedeutung
0:	Spur des nächsten Directory-Blocks,
1:	Nummer des nächsten Directory-Blocks,
2:	Dateityp. Relevant sind die Bits #0 bis 2, Bit #7 ist im Normalfall immer gesetzt (= $80): Das bedeutet, daß es sich um eine ordnungsgemäß geschlossene Datei handelt. Typkennzeichnungen: $80 = DEL, $81 = SEQ, $82 = PRG, $83 = USR, $84 = REL. Wird Bit #6 (Normalwert: 0) ebenfalls eingeschaltet, aktiviert man den Schreib- und Löschschutz für die jeweilige Datei.
3/4:	Spur- und Sektornummer des ersten Datenblocks dieses Files,
5 bis 20:	Dateiname (maximal 16 Zeichen), aufgefült mit CH$(160). Bei kurzen Dateinamen kann man noch Ladeanweisung anhängen (.z.B. »,8,1:«).
21/22:	nur REL-Files: Spur und Sektor des ersten Side-Sektor-Blocks.
23:	nur REL-Files: Datensatzlänge,
24 bis 27:	unbenutzt,
28/29:	Zwischenspeicher für REPLACE (SAVE@, mit Klammeraffe!),
30/31:	Blockanzahl, die von der Datei belegt werden (im Low-High-Byte-Format).
32 bis 63:	dito, nächster Directory-Eintrag,
...	
224 bis 255:	letzter Eintrag dieses Directory-Sektors, maximal acht sind möglich.

Tabelle 2. Directory-Sektor 1: Bedeutung der Bytes

SYNC
08
PRÜF-SUMME
SEKTOR
Spur
ID2
ID1
Lücke
SYNC
07
SPUR
Sektor
254 BYTE DATEN
PRÜF-SUMME
LÜCKE
HEADER
DATENBLOCK
Zeiger auf nächsten Sektor

Aufbau eines Diskettensektors (GCR-Format)

Jobcodes der Floppy 1541 (DOS 2.6)

Wert (hexadezimal/dezimal):	Funktion
$80 (128):	einen Block in den Puffer lesen
$90 (144):	einen Block aus dem Puffer schreiben
$A0 (160):	Sektor mit dem Pufferinhalt vergleichen (VERIFY)
$B0 (176):	Test, ob Sektor vorhanden
$C0 (192):	Bump des Schreib-Lesekopfs
$D0 (208):	Maschinenprogramm im Puffer ausführen
$E0 (224):	Programm im Puffer nach Einschalten des Laufwerks ausführen

Rückmeldungen der Jobschleife

Wert (hexadezimal):	Fehlerursache:	Fehlermeldung
$01:	kein Fehler:	00, ok
$02:	Blockheader nicht gefunden:	20, Read Error
$03:	SYNC-Markierung nicht gefunden:	21, Read Error
$04:	Datenblock nicht gefunden:	22, Read Error
$05:	falsche Datenprüfsumme:	23, Read Error
$07:	Fehler bei Verify:	25, Write Error
$08:	Diskette ist schreibgeschützt:	26, Write Protect on
$09:	falsche Prüfsumme im Blockheader:	27, Read Error
$0A:	zu langer Datenblock auf Disk:	28, Write Error
$0B:	falsche ID im Blockheader:	29, Disk ID mismatch
$0F:	keine Disk im Laufwerk oder nicht verriegelt:	74, Drive not ready
$10:	Fehler bei Decodierung:	24, Read Error

Tabelle 3. Kurz und bündig: 1-Byte-Befehle fürs Laufwerk (Jobcodes)

Floppy-Status-Flag $90

Bit-Nr. (Binärwert):	Fehler, wenn gesetzt (=1)
1 (2):	Zeitüberschreitung bei IEC-Eingabe
2 (4):	Zeit bei IEC-Ausgabe überschritten
3 bis 5 (8 bis 32):	nur für Datasette
6 (64):	Übertragung beendet
7 (128):	Gerät meldet sich nicht (74, Device not present). Identisch mit Jobcode-Rückmeldung $0F.

Tabelle 4. Fehlerursachen, die das Status-Byte beeinflussen

Drive, Disk-Drive: Diskettenlaufwerk.

Fehlerkanal: besser: Befehlskanal der Floppy. Wird beim Öffnen (OPEN-Befehl) durch die Sekundäradresse 15 eingestellt. Ermöglicht die Übergabe von Block- und Memory-Anweisungen ans Laufwerk oder die Analyse der Daten-Bytes, die z.B. einen Floppyfehler im Klartext ausgeben.

File: Fachausdruck für »Diskettendatei«. Der Name wird normalerweise auch in die BAM und ins Directory eingetragen. Ausnahme: Direktzugriffsdateien (s. dort).

Group Code Recording (GCR): speziell von Commodore-Floppies verwendetes Aufzeichnungsformat für den Magnetfluß beim Beschreiben oder Lesen einer Spur.

Header: Name und ID-Kennung einer Diskette. Beide Daten sind auf Spur 18, Sektor 0 abgelegt.

High Density (HD): hohe Dichte. Im Vergleich zu DD-Disketten (s. Double Density) wurde der magnetische Flußwechsel erhöht (jetzt: 135 tpi). Das geht nur durch Veränderung der Diskettenbeschichtung: Bei HD-Disketten ist sie dünner, aber härter. Man sollte solche Disketten nur mit geeigneten Laufwerken bearbeiten (z.B. mit dem eines IBM-kompatiblen AT), das bei 5¼-Zoll-Disketten 1,2 MByte und bei 3½-Zoll-Scheiben 1,44 MByte Speicherkapazität erreicht. Die Floppies 1541/1571 sind dafür nicht geeignet: Kaufen Sie lieber keine HD-Disketten – Sie könnten (wenn nicht schon beim Formatieren!) während des Speicherns wichtiger Programme oder Daten Probleme bekommen! Unproblematisch ist dagegen die 1581, wenn Sie statt einer DD eine HD-Disk zum Laden und Speichern verwenden.

Jobcodes: signifikante 1-Byte-Kommandos an den Disk-Controller, die man per DOS-Anweisung Memory-Write (M-W) abschickt. Die entsprechenden Speicherstellen liegen in der Floppy-Zeropage (Adressen $00 bis $05). Nach Ausführung der Aufgabe (Job) enthalten dieselben Speicherstellen die DC-Rückmeldungen. Tabelle 3 zeigt alle Jobcodes und deren Funktionen.

Label: Etikett auf der Vorderseite mit Infos zum Disketteninhalt oder Qualitätshinweise (Spur- oder Speicherdichte, nutzbare Seiten usw.).

Mini-Disk: Floppydiskette mit 5¼ Zoll Durchmesser (z.B. für die Laufwerke 1541, 1570 und 1571).

SYNC-Markierung: Jeder Sektor besteht aus Blockheader und dem entsprechenden Datenblock (Abb.). Die Synchron-(SYNC-)Markierungen bestehen aus mehreren $FF-Bytes. Findet der Schreib-Lesekopf solche Werte, kümmert er sich ums nächste Kennzeichen: Hat es den Wert $08, weiß der Disk-Controller, daß ein Blockheader folgt. Die Zahl $07 gibt an, daß nun ein 256 Byte großer Datenblock folgt. Nicht vergessen: 2 Byte muß man immer für die Spur- und Sektornummer als Zeiger auf den logisch folgenden Sektor abziehen!

Single Density: einfache Spurdichte,

Single Sided (SS, 1S): Diskette kann zwar auf eigenes Risiko beidseitig bespielt werden, beim Hersteller wurde aber nur die Vorderseite geprüft (s. Double Sided).

Speeder: Soft- oder Hardware-Erweiterungen des Floppy-Betriebssystems, um Datenzugriffe (Laden und Speichern) zu beschleunigen. Hardware-Speeder sind generell mit dem Umbau der Floppystation verbunden. Damit umgeht man die langsamere serielle Datenübertragung und stellt auf Parallelbetrieb um (per zusätzlichem Kabel, z.B. am Expansionport). Software-Speeder bestehen aus einem Erweiterungsmodul, das in den Expansionsport gesteckt wird (seltener aus einem Softwareprogramm, das man ins Computer-RAM lädt und startet, z.B. »RAM-Exos« im 64'er-Sonderheft 62). Bekannte Speeder: Jiffy DOS, Rex DOS, Prologic DOS classic, Profi DOS (s. 64'er-Magazin 11/92).

Status-Byte: vom Betriebssystem benutzte Speicherstelle $90 (144) in der Zeropage des C64. Der Normalwert ist 0. Jedes gesetzte Bit signalisiert einen Störfall, der beim Floppybetrieb oder bei der Datenübertragung (z.B. SAVE) auftritt (s. Tabelle 4). Achtung: Die Fehler sind, bis auf den von Bit 7 verursachten, nicht identisch mit den Jobcode-Rückmeldungen des Disk-Controllers (s. Jobcode).

tpi: Abkürzung für Tracks per Inch (Spuren pro Zoll); die Spurdichte einer Diskette (auf den Disk-Labels vermerkt).

Floppysteuerung und -programmierung in Assembler halten viele für eine schwer zu knackende Nuß. Mitnichten! Wir zeigen Ihnen, wie leicht man simple Lade- und Speicherroutinen, aber auch spezielle Floppybefehle handhabt. Assembler-Grundkenntnisse müssen wir allerdings voraussetzen (notfalls kann man im 64'er-Sonderheft 71 nachschlagen!). Aber keine Angst: Statt Fach-Chinesisch zu verbreiten, geht's uns in erster Linie darum, zu zeigen, wie man bereits im Betriebssystem des Computers vorhandene Routinen sinnvoll einsetzt.

Zunächst verschaffen wir uns Klarheit über alle Sprungbefehle im Kernel-ROM (ab $E000), die man zur Floppy-Programmierung braucht (Tabelle 2). In unseren Listingbeispielen werden Sie die Routinen-Kürzel stets wiederentdecken.

Welche Aufgaben erfüllen die Betriebssystemroutinen?

SETLFS und SETNAM

...sind für alle weiteren Befehle enorm wichtig! Bitte erst gar nicht weiterlesen, wenn Sie nicht sicher sind, den Sinn der beiden Anweisungen verstanden zu haben!

Bei der Verwendung von Basic-Befehlen sind wir gewöhnt, wichtige Parameter direkt mit dem Befehl anzugeben (z.B. den File-Namen beim Laden oder die Werte zum Öffnen einer Datei):

```
LOAD "Programm",8
```

oder

```
OPEN 2,8,2,"Datei,S,R"
```

In Assembler sind diese beiden Unterroutinen für die Parameterzuweisung verantwortlich:

SETLFS ($FFBA): Dateinummer, Geräte- und Sekundäradresse definieren. Achtung: Beachten Sie die vorgeschriebenen Aufgaben des Akkumulators A sowie des x- und y-Registers. Sie sind nicht austauschbar! Beispiel:

```
LDA #lf ;lf = logische Filenummer
LDX #ga ;ga = Geräteadresse
LDY #sa + $60 ;sa = Sekundäradresse
JSR $FFBA ;Parameter an SETLFS übergeben
```

Bei Befehlen wie OPEN oder LOAD muß man dem Laufwerk mitteilen, welches File wo bearbeitet werden soll. Dazu dient die Routine SETLFS (Datei-Spezifikationen setzen). Die **logische Dateinummer** (muß in den Akku!) darf Werte von 1 bis 127 erhalten. Damit kann man z.B. Daten speziell für eine Anweisung nachschieben. Beachten Sie, daß maximal zehn geöffnete Dateien vom Betriebssystem verwaltet werden!

Die **Geräteadresse** wird im x-Register abgelegt. Die Werte bedeuten:
- 0: Tastatur,
- 1: Datasette,
- 2: RS232-Schnittstelle (z.B. bei DFÜ),
- 3: Bildschirm,
- 4 und 5: Drucker,
- 6 und 7: nicht belegt. Mit Nr. 6 wurde früher der Commodore-Plotter deklariert; den gibt's aber heute nicht mehr.
- 8 bis 11: Floppy.

Für die **Sekundäradresse** (muß im y-Register stehen!) kann man Werte von 0 bis 15 wählen. Drei sind jedoch für feste Funktionen reserviert:
- 0: Laden,
- 1: Speichern,
- 15: Befehlkanal der Floppy öffnen.

Vor der Übergabe der Sekundäradresse sollte man den Wert $60 (96) addieren.

SETNAM ($FFBD): Länge und Adresse des Dateinamens im Speicher übergeben. Beispiel:

```
LDA #fn ;fn = Länge des Filenamen
LDX #<name ;name = Adresse des
LDY #>name ;Filenamens
JSR  $FFBD
NAME .text "beispiel"
```

Geben Sie die Anzahl der zu übertragenden Zeichen (=Länge des Dateinamens) im Akku an und die Startadresse (wo die Bytes abgelegt sind – irgendwo im freien Computer-RAM, also auch im Variablenspeicher) in den Registern X (Low-Byte) und Y (High-Byte).

LOAD und VERIFY ($FFD5)

Beginnen wir mit einem einfachen Beispiel: eine Laderoutine, die eine Datei »TEST« in den Speicher ab $C000 (49152) holt:

```
SETLFS = $FFBA
SETNAM = $FFBD
```

Grundlagen der Floppy-Programmierung

Viele Spuren führen zum Ziel

Von (A)ssembler bis (B)asic: Die intelligenten Laufwerke von Commodore befolgen jede Anweisung des C 64 – man muß sich nur verständlich ausdrücken!



```
LDX #$08 ;ga = 8 (Floppy)
LDY #$00 ;sa = 0: relatives laden
JSR SETLFS ;sa = 1: absolutes laden
LDA #$04 ;Übergabe des
LDX #<name ;Filenamens
LDY #>name
JSR SETNAM
LDA #$00 ;0 = LOAD / 1 = VERIFY
LDX #<$C000 ;Ladeadresse Low-Byte ($00)
LDY #>$C000 ;Ladeadresse High-Byte ($C0)
JSR $FFD5
```



```
RTS
...
NAME .text "test"
```

Beim Laden ist die logische Dateinummer bedeutungslos und kann entfallen. Die Sekundäradresse gibt an, ob die Daten relativ oder absolut geladen werden sollen. Übergibt man den Wert 0 ans y-Register, darf man eine beliebige Ladeadresse angeben. »1« übernimmt die beim Sichern festgesetzte Speicheradresse – die Angabe einer Ladeadresse ist also überflüssig. Wenn man aber einen bestimmten Speicherbereich definiert, in den man laden will (sa = 0), muß die Adresse vor dem Aufruf der LOAD-Routine als Low-High-Byte im x- und y-Register abgelegt werden!

Da $FFD5 auch die Verify-Routine angesprungt, muß zur Unterscheidung »0« (Laden) oder »1« (Vergleichen) im Akku stehen.

Statusregister $90

Entsteht während der Arbeit mit der Routine ein Fehler, läßt er sich per Statusregisters $90 (144) identifizieren. Immer, wenn der Inhalt dieser Speicherzelle nicht $40 (64) ist, hat sich mit tödlicher Sicherheit ein Diskfehler eingeschlichen! Hier die Assembler-Zeilen zur Abfrage des Statusregisters, die man als Unterprogramm ablegen und per JSR angespringen sollte:

```
LDA $90 ;Statusregister in Akku
CMP #%01000000 ;Vergleich mit $40
BNE error
RTS
error: (Reaktion auf Diskfehler, z.B.
selbstdefinierte Meldung usw.)...
```

Bit #6 (= $40) ist stets gesetzt, wenn die Datenübertragung von oder zur Floppy abgeschlossen ist. Die einzelnen Bits des Registers $90 und ihre Bedeutung:

- Bit 0 (1): Fehler beim Schreiben,
- Bit 1 (2): Fehler beim Lesen,
- Bit 6 (64): Datenende erreicht,
- Bit 7 (128):Gerät meldet sich nicht (Einschalten oder Anschließen vergessen?)

SAVE ($FFD8)

Speichern von Datenbereichen ist nur unwesentlich komplizierter als das Laden. Unser Beispielprogramm sichert den aktuellen Bildschirm im Textmodus ($0400 (1024) bis $07E7 (2023):

```
SETLFS = $FFBA
SETNAM = $FFBD
SAVE = $FFD8
LDX #$08 ;Geräteadresse
JSR SETLFS ;angeben
LDA #$06 ;Übergabe des
LDX #<name ;Filenamen
LDY #>name
JSR SETNAM
LDA #$FB ;Zeiger auf Ablage der Startadresse
LDX #<$0400 ;Startadresse Low-Byte
LDY #>$0400 ;Startadresse High-Byte
STX $FB ;Ablage der Startadresse im Low-High-
STY $FC ;Byte-Format ab $FB
LDX #<$07E7 + 1 ;Endadresse +1 Low-Byte
LDY #>$07E7 + 1 ;Endadresse +1 High-Byte
JSR SAVE
RTS
...
name .text "screen"
```

Beim Speichern ist die Sekundäradresse bedeutungslos. Die Übergabe von Geräteadresse und Dateinamen (welche Länge, wo abgelegt) bereiten uns ebenfalls keine Probleme mehr. Unübersichtlicher wird's allerdings bei der Parameter-Definition von Start- und Endadresse. Da wir lediglich drei Register haben (A, X u. Y), aber vier Werte angeben müßten, ha-

Forsetzung auf Seite 22

So finden Sie die Programme auf der Diskette

DISKETTE SEITE 1

Blöcke	Name	Typ	Seite
Ø	"64'ER SONDERH.83" S1 2A		
29	"DISKLADER"	PRG	S. 21
Ø	" ┌────────────┐ "	USR	
Ø	" │ TOOLS │ "	USR	
Ø	" └────────────┘ "	USR	
37	"ULTRAMENUE MAKER"	PRG	S. 30
28	"ULTRABOOT MAKER+"	PRG	
4	"ULTRAMENU TOOL"	PRG	
9	"ULTRALOAD PLUS"	PRG	
Ø	"----------------"	USR	
91	"C64-TOOLS"	PRG	S. 36
1	"INFOTEXT"	SEQ	
8	"TESTDATEI 168"	PRG	
4	"TESTDATEI 165"	SEQ	
1Ø	"TESTDATEI 128"	USR	
41	"RANDOMFILES"	PRG	
Ø	"----------------"	USR	
16	"SIR-SCAN"	PRG	S. 35
9	"I-SIRSCAN"	PRG	
Ø	"----------------"	USR	
2Ø	"DISC-SCANNER 4Ø"	PRG	S. 32
Ø	"----------------"	USR	
5	"DISK-SEARCHER"	PRG	S. 34
Ø	"----------------"	USR	
23	"ULTRAFORMAT V2"	PRG	S. 40
1	"DISKSWITCH CØØØ"	PRG	
1	"DISKSWITCH Ø33C"	PRG	
2	"SWITCH U-O.BAS"	PRG	
1	"SWITCH O-U.BAS"	PRG	
Ø	"----------------"	USR	
5	"FAST FORMAT"	PRG	S. 35
5	"FF.OBJ"	PRG	
Ø	"----------------"	USR	
7	"DOUBLE-FORMAT"	PRG	S. 41
Ø	" ┌────────────┐ "	USR	
Ø	" │ KOPIERPRG. │ "	USR	
Ø	" └────────────┘ "	USR	
18	"TRACK-COPY II"	PRG	S. 41
Ø	"----------------"	USR	
12	"SUPERCOPY"	PRG	S. 44
Ø	"----------------"	USR	
32	"WIZCOPY V1.Ø"	PRG	S. 42
1Ø	"WIZCOPY-EDITOR"	PRG	
Ø	" ┌────────────┐ "	USR	
Ø	" │ FLOPPY 1581 │ "	USR	
Ø	" └────────────┘ "	USR	
12	"DUAL-FILECOPY"	PRG	S. 9
Ø	"----------------"	USR	
2	"PRG 1.BAS"	PRG	S. 4
2	"PRG 2.BAS"	PRG	
3	"PRG 3.BAS"	PRG	
Ø	"----------------"	USR	
3	"SUBDIR.1581"	PRG	
Ø	" ┌────────────┐ "	USR	
Ø	" │ DISKETTE │ "	USR	
Ø	" │ BEIDSEITIG │ "	USR	
Ø	" │ BESPIELT │ "	USR	
Ø	" └────────────┘ "	USR	

DISKETTE SEITE 2

Blöcke	Name	Typ	Seite
Ø	"64'ER SONDERH.83" S2 2A		
29	"DISKLADER"	PRG	S. 21
Ø	" ┌────────────┐ "	USR	
Ø	" │ TIPS & TRICKS │ "	USR	ab S. 45
Ø	" └────────────┘ "	USR	
2	"IRQ-DOS $CEØØ"	PRG	
Ø	"----------------"	USR	
11	"FLOADUTIL"	PRG	
6	"I-FLOADUTIL"	PRG	
Ø	"----------------"	USR	
7	"FORMAT 1-41"	PRG	S. 28
6	"FORMAT 36-41"	PRG	
5	"TRACKWRITE"	PRG	
4	"TRACKREAD"	PRG	
1	"FP.FORMATER.OBJ"	PRG	
Ø	"----------------"	USR	
6	"FLOPROM-READER"	PRG	
Ø	"----------------"	USR	
13	"FILE REVIVAL"	PRG	
Ø	"----------------"	USR	
3	"PHYSICAL SCRATCH"	PRG	
Ø	" ┌────────────┐ "	USR	
Ø	" │ GRUNDLAGEN │ "	USR	
Ø	" └────────────┘ "	USR	
14	"EDDI"	PRG	ab S. 16
3	"5-BYTE-ID"	PRG	
3	"NEW HEADER"	PRG	
3	"READ DIR"	PRG	
3	"WRITE DIR"	PRG	
8	"NO SCRATCH"	PRG	
9	"DIR-SORTER"	PRG	
2	"DISK PROTECT"	PRG	
2	"LED-BLINKER"	PRG	
1	"SOFT FLOPPY"	PRG	
3	"READ-ERROR 22"	PRG	
3	"READ-ERROR 23"	PRG	
Ø	" ┌────────────┐ "	USR	
Ø	" │ ENDE │ "	USR	
Ø	" └────────────┘ "	USR	

WICHTIGE HINWEISE

zur beiliegenden Diskette:

Aus den Erfahrungen der bisherigen Sonderhefte mit Diskette, wollen wir ein paar Tips an Sie weitergeben:

1

Bevor Sie mit den Programmen auf der Diskette arbeiten, sollten Sie unbedingt eine Sicherheitskopie der Diskette anlegen. Verwenden Sie dazu ein beliebiges Kopierprogramm, das eine komplette Diskettenseite dupliziert.

2

Auf der Originaldiskette ist wegen der umfangreichen Programme nur wenig Speicherplatz frei. Dies führt bei den Anwendungen, die Daten auf die Diskette speichern, zu Speicherplatzproblemen. Kopieren Sie daher das Programm, mit dem Sie arbeiten wollen, mit einem File-Copy-Programm auf eine leere, formatierte Diskette und nutzen Sie diese als Arbeitsdiskette.

3

Die Rückseite der Originaldiskette ist schreibgeschützt. Wenn Sie auf dieser Seite speichern wollen, müssen Sie vorher mit einem Diskettenlocher eine Kerbe an der linken oberen Seite der Diskette anbringen, um den Schreibschutz zu entfernen. Probleme lassen sich von vornherein vermeiden, wenn Sie die Hinweise unter Punkt 2 beachten.

ALLE PROGRAMME
aus diesem Heft



HIER

Chefredakteur: Georg Klinge (gk) – verantwortlich für den redaktionellen Teil
Stellv. Chefredakteur: Arnd Wängler (aw)
Textchef: Jens Maasberg
Redaktion: Harald Beiler (bl), Herbert Großer (gr)
Producer: Andrea Pfliegensdörfer
Redaktionsassistenz: Birgit Misera, Helga Weber

So erreichen Sie die Redaktion:
Tel. 089/4613-202, Telefax: 089/4613-5001, Btx: 64064

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, so muß das angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in den von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträgern. Mit Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß die Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Layout: Dagmar Portugall
Bildredaktion: Wallo Linne (Ltg.), Roland Müller, Tina Steiner (Fotografie)
Titelgestaltung und -grafik: Wolfgang Berns

Anzeigenleitung: Peter Kusterer
Anzeigenverwaltung und Disposition: Christopher Mark (421)

Anzeigen-Auslandsvertretung:
Großbritannien und Irland: Smyth International, Telefon 0044/81340-5058, Telefax 0044/81341-9602
Niederlande und Belgien: Insight Media, Telefon 0031/2153120 42, Telefax 0031/21531 0572
Italien: Medias International, Telefon 0039/3175 1494, Telefax 0039/3175 1482
USA und Kanada: M & T International Marketing, Telefon 001/415358-9500, Telefax 001/415358-9739
Japan: Media Sales Japan, Telefon 0081/33504-1925, Telefax 0081/33595-1709
Taiwan: Acer TWP Corporation, Telefon 00886-2-7136959 Telefax 00886-2-7 151950
Korea: Young Media Inc., Telefon 0082-2-7 564819, Telefax 0082-2-7575789
Israel: Baruch Schaefer, Telefon 00972-3-5562256, Telefax 00972-3-5566944
International Business Manager: Stefan Grajer 089/4613-638

So erreichen Sie die Anzeigenabteilung:
Tel. 089/4613-962, Telefax: 089/4613-791

Gesamtvertriebsleiter: Helmut Grünfeldt
Leiter Vertriebsmarketing: Benno Gaab

Vertrieb Handel: MZV, Moderner Zeitschriften Vertrieb GmbH & Co. KG, Breslauer Straße 5, Postfach 11 23, 8057 Eching, Tel. 089/319006-0

Verkaufspreis: Das Einzelheft kostet DM 16,-

Produktion: Klaus Buck (Ltg./180), Wolfgang Meyer (Stellv./887)

Druck: SOV. Graphische Betriebe, Laubanger 23, 8600 Bamberg

Haftung: Für den Fall, daß im 64'er Sonderheft unzutreffende Informationen oder in veröffentlichten Programmen oder Schaltungen Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlags oder seiner Mitarbeiter in Betracht.

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken erhältlich. Anfragen an Klaus Buck, Tel. 089/4613-180, Telefax 089/4613-232

Vorstand: Dr. Rainer Doll, Lutz Glandt, Dieter Streit

Verlagsleitung: Wolfram Höfler
Operation Manager: Michael Koeppe

Direktor Zeitschriften: Michael M. Pauly

Anschrift des Verlags: Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

ISSN 0931-8933

Die Zeitschrift wird mit chlorfreiem Papier hergestellt.

Copyright-Erklärung

Name:..

....

Anschrift:..

Datum:..

Computertyp:..

Benötigte Erweiterung/Peripherie:..

Datenträger: Kassette/Diskette..

Programmart:..

Ich habe das 18. Lebensjahr bereits vollendet

...., den ....

....

(Unterschrift)

Wir geben diese Erklärung für unser minderjähriges Kind als dessen gesetzliche Vertreter ab.

...., den ....

....

Bankverbindung:

Bank/Postgiroamt:..

Bankleitzahl:..

Konto-Nummer:..

Inhaber des Kontos:..

Das Programm/die Bauanleitung:..

das/die ich der Redaktion der Zei[illegible] 64'er übersandt habe, habe ich selbst erarbeitet und nic[illegible] nicht teilweise, anderen Veröffentlichungen entnomme[illegible] Programm/die Bauanleitung ist daher frei von Rechten an[illegible] und liegt zur Zeit keinem anderen Verlag zur Veröffentlichung v[illegible] Ich bin damit einverstanden, daß die Markt & Technik Verlag A[illegible] das Programm/die Bauanleitung in ihren Zeitschriften oder i[illegible]en herausgegebenen Büchern abdruckt und das Programm/die Bauanleitung vervielfältigt, wie beispielsweise durch Herstellung von Disketten, auf denen das Programm gespeichert ist, oder daß sie Geräte und Bauelemente nach der Bauanleitung herstellen läßt und vertreibt bzw. durch Dritte vertreiben läßt.

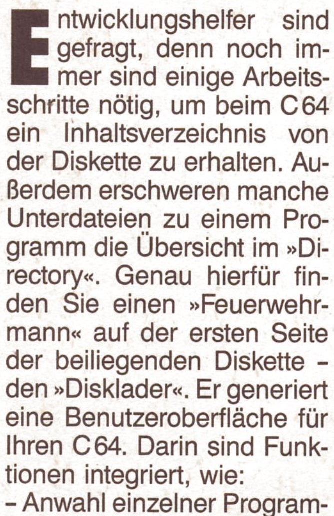
Ich erhalte, wenn die Markt & Technik Verlag AG das Programm/die Bauanleitung druckt oder sonst verwertet, ein Pauschalhonorar.

Disklader – Programme laden mit Komfort

Diskettenoberfläche de Luxe

Keine umständlichen Ladeanweisungen und ein übersichtliches Inhaltsverzeichnis der Diskette auf dem Bildschirm. Unser »Disklader« erfüllt auch gehobene Ansprüche.

von Herbert Großer



Entwicklungshelfer sind gefragt, denn noch immer sind einige Arbeitsschritte nötig, um beim C64 ein Inhaltsverzeichnis von der Diskette zu erhalten. Außerdem erschweren manche Unterdateien zu einem Programm die Übersicht im »Directory«. Genau hierfür finden Sie einen »Feuerwehrmann« auf der ersten Seite der beiliegenden Diskette – den »Disklader«. Er generiert eine Benutzeroberfläche für Ihren C64. Darin sind Funktionen integriert, wie:

- Anwahl einzelner Programme (mit jeweiliger Kurzbeschreibung),
- automatisches Laden und Starten von Diskette oder
- Erkennung der richtigen Diskette bzw. Diskettenseite.

Da sich der Disklader an erster Stelle auf der Diskette zum Sonderheft befindet, genügt es, zum Laden einzugeben:

```
LOAD":*",8
```

Nach der Bestätigung mit <RETURN> dauert es ca. 15 s, bis die Datei im Speicher ist. Sie starten mit RUN und <RETURN>. Anschließend wird das File entpackt (ca.2 s) und es erscheint die Benutzeroberfläche des »Disklader« (s. Abbildung). In der rechten unteren Bildschirmhälfte sehen Sie weiß umrandet den Namen des ausgewählten Programms. Die unterste Bildschirmzeile ist die dazugehörige Kurzerklärung. Zusätzlich finden Sie in der rechten unteren Bildschirmhälfte den Text »Seite 1 auf Disk« oder »Seite 2 auf Disk«. Da Sie die Inhaltsverzeichnisse beider Seiten (ohne die Disk zu wenden) durchblättern können, finden Sie hier den Hinweis, auf welcher Diskettenseite sich das gewählte Programm befindet.

angewähltes programm — Kurzerklärung für das angewählte Programm — Seite auf der Diskette und Feld für Fehlermeldungen

Durch Tastendruck <CRSR aufwärts> bzw. <CRSR abwärts> wählen Sie das nächste oder vorherige Programm. Sie blättern quasi durch den Inhalt der Programme. <HOME> bringt Sie zum ersten Eintrag des Inhaltsverzeichnisses. Selbstverständlich sind nur die Programme verzeichnet, die sich eigenständig laden oder starten lassen.

<RETURN> führt Sie in den Ladeteil. Ist kein Diskettenfehler aufgetreten, erscheint kurzzeitig »00,OK, 00,00« am Bildschirm. Eventuelle Fehleranzeigen bleiben sichtbar am Bildschirm (z.B. »21,READ ERROR, 18,00«= Drive not ready). Sie lassen sich durch einen beliebigen Tastendruck wieder löschen. Schlagen Sie bitte vorher im Handbuch Ihrer Floppy nach und beseitigen Sie den Fehler. Eine andere Art der Fehlermeldung wird durch einen blinkenden Text dargestellt (z.B. »Bitte Disk wenden« oder »Falsche Diskette«). Sind Fehler ausgeblieben, lädt der Disklader das von Ihnen gewählte Programm von der Diskette und startet es. Ladefehler, die in dieser Phase auftreten, werden nicht mehr berücksichtigt: Der Disklader wird vom neuen Programm einfach überschrieben. Sonst könnten wir nur Programme veröffentlichen, die mit der Benutzeroberfläche zusammenarbeiten. Bei vielen Spielen, Tricks oder Tools ist dies aber nicht der Fall.

Für Sie bedeutet dies, nach jedem Starten eines Programms den »Disklader« erneut zu laden. Wer die Benutzeroberfläche verlassen will, gibt <RUN/STOP> ein. Sie befinden sich dann im normalen »Basic« des C64. Für einen Neustart befehlen Sie

```
SYS 12032
```

und bestätigen mit <RETURN>. Dieser Neustart funktioniert auch nach einem Reset, d.h. wenn Sie durch den entsprechenden Taster einen Hardware-Reset ausgelöst haben. Allerdings sollten Sie zwischenzeitlich kein Programm geladen haben, da dies den verwendeten Speicherbereich überschreiben könnte. Laden Sie in diesem Falle den Disklader neu.

Wir haben bei der Programmierung größten Wert auf Kompatibilität mit den unterschiedlichsten Betriebssystemerweiterungen gelegt. Lediglich bei der Gerätekonfiguration C128 mit RAM-Erweiterung und zweiter Diskettenstation sollten Sie die externe Floppy ausschalten. (gr)

Kurzinfo: Disklader

Programmart: Hilfsprogramm zum Laden der Programme auf der beiliegenden Diskette
Laden: LOAD":*",8
Starten: nach dem Laden mit RUN
Steuerung: Tastatur
Programmautor: H. Großer

ben sich die Entwickler des C-64-Betriebssystems einen Trick einfallen lassen: Die Startadresse wird als Low-High-Byte in zwei freien Zeropage-Adressen (zwischen $00 und $FF) abgelegt und im Akku ein entsprechender Zeiger gesetzt (Adresse des Low-Bytes). Hier bieten sich die Adressen $FB (251) bis $FE (254)an, da sie vom Betriebssystem nie für eigene Zwecke gebraucht werden. Die Endadresse des zu speichernden Bereichs muß um »1« erhöht und in den Registern X und Y eingetragen werden.

Das Speichern beeinflußt das Carry-Bit: Ist ein Fehler aufgetreten, wird es gesetzt. So könnte die Unterroutine zur Fehlerabfrage aussehen:

```
JSR SAVE
BCS fehler
RTS
fehler ...(Meldung usw.)
```

Bisher war's leicht: Der Umgang mit einfachen Lade- und Speicherroutinen ist Ihnen nun bestimmt in Fleisch und Blut übergegangen. Wir können uns also getrost in etwas schwierigere Gefilde wagen.

Spezielle Floppybefehle

Da die Floppy 1541 (und ihre Schwestern 1570, 1571 und 1581) mit eigenem Mikroprozessor arbeiten, eröffnet sich Programmierern ein großer Tummelplatz. Ohne Probleme kann man auf den Befehlssatz (ROM) des Diskettenlaufwerkes zugreifen, um Daten noch effektiver zu manipulieren.

Wie in Basic muß man zunächst den Befehlskanal öffnen und die gewünschte Anweisung an die Floppy senden. Nicht vergessen: Kanal wieder schließen!

OPEN ($FFC0) – CLOSE ($FFC3) – CLALL ($FFE7)

Der simple Floppy-Befehl zum Formatieren einer Diskette:

```
OPEN 15,8,15,"N:TESTDISK,01": CLOSE15
```

sieht als Assembler-Listing so aus:

```
100 ;*** Befehlübermittlung ***
101 SETLFS = $FFBA
102 SETNAM = $FFBD
103 OPEN = $FFC0
104 CLOSE = $FFC3
105 CLALL = $FFE7
110 LDA #$0F ;Filenummer
120 LDX #$08 ;Geräteadresse
130 LDY #$6F ;sa=15: Befehlskanal + $60
140 JSR SETLFS
150 LDA #$0D ;Länge und Übergabe der
160 LDX #<name ;Befehlssequenz
170 LDY #>name
180 JSR SETNAM
190 JSR OPEN ;öffnen und übertragen
200 LDA #$0F ;Dateinummer 15 wieder
210 JSR CLOSE ;schließen
220 RTS
230 ;*** Floppybefehl ***
240 name .text "n:testdisk,01"
```

Denken Sie dran: Floppyanweisungen immer mit Sekundäradresse 15 übermitteln! Nur so kapiert das Laufwerk, daß jetzt ein Befehl kommt! Der Befehlstext für die Floppy wird genauso übergeben, wie der Dateiname in den Beispielen davor. Zum Aktivieren der OPEN-Anweisung braucht man keine Parameter mehr (die haben Sie schon mit SETLFS und SETNAM definiert!). Beim Schließen muß man aber nochmals die File-Nummer nennen. Wer alle Dateiparameter auf einmal löschen will, sollte statt dessen ab Listingzeile 200 die CLALL-Routine (Close All) eintragen (dann wird die Routine sogar um einige Byte kürzer!):

```
200 JSR CLALL
210 RTS
```

Mit dem Floppybefehl »S:« (Scratch) wollen wir das Programm zum zweiten Mal testen. Angenommen, Sie möchten die Datei »Screen«, die wir per »Save«-Routine erzeugt haben, wieder von der Diskette verbannen. Dann sind folgende Zeilen zu ändern:

```
150 LDA #$08
240 name .text "s:screen"
```

Nach dieser Methode kann man auch Disketten initialisieren (i0:), validieren (v0:), Dateien kopieren (c:name) oder umbenennen (r:neuname=altname).

SAVE und LOAD für Profis

Bei vielen Anwendungen kommt es vor, daß die zu speichernden Daten nicht in einem Block zusammengefaßt sind, sondern an verschiedenen Stellen im RAM liegen. Ideales Beispiel: Hires-Grafikdateien. Die Bitmap liegt getrennt vom Video-RAM, den Farbspeicher muß man wieder in ganz anderen Regionen suchen. Da es grobe Verschwendung wertvollen Speicherplatzes wäre, diese Daten in einem Block zu speichern (das ergäbe eine Superdatei mit mehr als 50 000 Byte!), zeigen wir Ihnen eine elegantere Lösung. Unser Assembler-Beispiel speichert jeweils die ersten 256 Byte aus dem Video-RAM ab $0400 (1024) und dem Farb-RAM ab $D800 (55296) mit dem Dateinamen »Test«:

```
SETLFS = $FFBA
SETNAM = $FFBD
OPEN = $FFC0
LISTEN = $FFB1
SECOND = $FF93
CIOUT = $FFA8
UNLSN = $FFAE
CLOSE = $FFC3
LDA #$01 ;Filenummer
LDX #$08 ;Geräteadresse
LDY #$62 ;Sekundäradresse
JSR SETFLS
LDA #$08 ;Übergabe des
LDX #<name ;Filenamens
LDY #>name
JSR SETNAM
JSR OPEN
LDA #$08 ;Geräteadresse
JSR LISTEN ;Floppy auf Empfang
LDA #$62 ;Sekundäradresse
JSR SECOND ;File ansprechen
LDA #$00 ;Startadresse Low-Byte
JSR CIOUT ;an Floppy senden
LDA #$04 ;Startadresse High-Byte
JSR CIOUT ;an Floppy senden
LDX #$00 ;256 Byte aus
label1: LDA $0400,X ;dem Video-RAM
JSR CIOUT ;an die Floppy
INX ;senden
BNE label1
LDX #$00 ;256 Byte aus
label2: LDA $D800,X ;dem Farb-RAM
JSR CIOUT ;an die Floppy
INX ;senden
BNE label2
LDA #$08 ;Empfang-Anweisung
JSR UNLSN ;zurücknehmen
LDA #$01 ;File
JSR CLOSE ;schließen
RTS
...
name: .text "test,p,w"
```

Per Anhängsel »,p,w« legt man beim OPEN-Befehl fest, daß man eine Programmdatei (Typ PRG) öffnen und erzeugen will. Sie haben die Wahl unter vier Dateitypen:



- P: Programm,
- R: Relativ,
- S: Sequentiell,
- U: User.

PRG-Dateien werden bei der Ablage auf Diskette immer mit der Startadresse (Low-High-Byte) als erste Bytes nach dem Blockverbinder abgelegt - unverzichtbar fürs spätere Laden. Also: Beim Speichern nicht vergessen, diese Bytes anzugeben! Der zweite Parameter signalisiert, ob geladen (,r = read) oder gespeichert (,w = write) werden soll.

LISTEN – SECOND

Jetzt muß man der Floppy mitteilen, daß sie Daten empfangen muß, ebenso, über welche Datei. Dafür stehen uns die Routinen LISTEN und SECOND zur Verfügung. Bevor Sie also mit einer Datenübertragung beginnen, sollten Sie das Laufwerk stets darüber informieren, ob es Empfänger oder Sender sein soll. Zusätzlich ist anzugeben, welche Datei angesprochen wird. Sie erinnern sich: Es können nämlich mehrere Datenkanäle gleichzeitig geöffnet sein (maximal zehn!).

Nach diesen Vorbereitungen beginnt man mit dem Datentransfer. Die zu übertragenden Bytes werden nacheinander in den Akkumulator geladen und über Adresse $FFA8 (CIOUT) abgeschickt. Nach Transfer-Ende wird die Empfangsanforderung zurückgenommen und das File wieder geschlossen.

Spuren- und Sektorenaufteilung einer 5¼-Zoll-Diskette (Commodore-GCR-Format)

Tracknr.	Anzahl der Sektoren	Byte pro Track
1 bis 17	21	7692
18 bis 24	19	7143
25 bis 30	18	6667
31 bis 35	17	6250

Tabelle 1

TALK – UNTALK

Nun zum Gegenteil: Wir holen die Daten wieder an die Quellposition im C-64-RAM zurück. Diesmal öffnet man eine Datei mit dem Zusatz »,r« (File lesen).

```
name .text "test,p,r"
```

und steigt nach dem OPEN-Befehl ein:

```
TALK = $FFB4
TKSA = $FF96
ACPTR = $FFA5
UNTALK = $FFAB
CLOSE = $FFC3
LDA #$08 ;Geräteadresse
JSR TALK ;Floppy auf Senden einstellen
LDA #$61 ;Sekundäradresse
JSR TKSA ;File ansprechen
JSR ACPTR ;Startadresse Low- und
JSR ACPTR ;High Byte überlesen
LDX #$00 ;256 Byte lesen
label1:JSR ACPTR ;und ins Video-RAM
STA $0400,X ;schreiben
INX
BNE label1
LDX #$00 ;256 Byte lesen
label2: JSR ACPTR ;und ins Farb-RAM
STA $D800,X ;schreiben
INX
BNE label1
LDA #$08 ;Anweisung an Sender (Floppy)
JSR UNTALK ;zurücknehmen
LDA #$01 ;File
JSR CLOSE ;schließen
RTS
```

Durch die diversen Anhängsel (Suffixe) für Filetyp und Modus lassen sich sehr leicht flexible Speicher- und Laderoutinen programmieren. Wichtig ist, daß man die Floppy auf Empfang oder Senden einstellt!

Auf Sektoren zugreifen: Blockbefehle

Interne Floppyabläufe organisiert das integrierte Betriebssystem (DOS, Disk Operation System). Es steuert alle Operationen, die beim Verwalten von Dateien und Programmen in Frage kommen. Außerdem erlaubt es uns, daß wir direkt auf DOS-Funktionen zugreifen und das Laufwerk noch effizienter nutzen können. Voraussetzung: grundlegende Kenntnisse über Aufbau und Einteilung von 5¼-Zoll-Disketten.

Die runde Magnetscheibe wird in 35 Spuren (Tracks) eingeteilt. Auf jeder befindet sich eine unterschiedliche Anzahl von Sektoren bzw. Blöcken, die exakt 256 Byte speichern. Da der Umfang der äußeren Spuren (1 bis 17) größer ist, bestehen sie aus mehr Blocks (21). Je höher die Spurnummer, desto geringer die Sektorenanzahl! Die Speicherkapazität pro Spur nimmt automatisch ab, je weiter man sich zu den inneren Spuren rund ums Scheibenloch bewegt. Tabelle 1 bringt eine Übersicht der Spuren- und Sektorenzahlen einer 5¼-Zoll-Disk im GCR-Format (Group Code Recording).

Um die Blockbefehle zu nutzen, muß ein Puffer im Floppy-RAM (also nicht im Computer!) reserviert werden. Diese Speicherbereiche sind stets 256 Byte groß: Der Inhalt eines Sektors paßt exakt hinein! Es gibt fünf Floppypuffer, Nr. 3 und 4 braucht die Floppy aber selbst: Directory bzw. die BAM (Block-Belegungs-Tabelle) sind dort abgelegt. Sie haben nun die Möglichkeit, einen anderen Bereich anzugeben oder der Floppy die Auswahl zu überlassen. Wir empfehlen die zweite Variante: So vermeidet man, daß der gewählte Pufferbereich belegt ist und der Direktzugriffsversuch wird mit einer Fehlermeldung abgebrochen.

Systemroutinen des C-64 zur Floppyprogrammierung

Bezeichnung	Adresse
SETLFS (FILPAR)	$ffba
SETNAM	$ffbd
LOAD	$ffd5
SAVE	$ffd8
OPEN	$ffc0
CLOSE	$ffc3
CLALL	$ffe7
LISTEN	$ffb1
SECOND (SECLST)	$ff93
UNLSN (UNLISTEN)	$ffae
CIOUT (IECOUT)	$ffa8
TALK	$ffb4
TKSA (SECTLK)	$ff96
UNTALK	$ffab
ACPTR (IECIN)	$ffa5
CKOUT	$ffc9
CHKIN	$ffc6
STROUT	$ab1e
CLRCHN (CLRCH)	$ffcc
BSOUT (PRINT)	$ffd2
BASIN	$ffcf

Tabelle 2

Bevor wir dazu übergehen, Daten auf Diskette mit Blockbefehlen zu manipulieren, sollten Sie eine Übungsdiskette ins Laufwerk legen (kopieren Sie irgendeine vollbeschriebene Diskette aus Ihrer Sammlung!). Die Gefahr ist nämlich groß, durch Fehler oder falsche Eingaben die Disk abzuschießen und wichtige Daten zu verlieren! Wenn Sie dann später mit Originaldaten jonglieren, sollten Sie sich vorher vergewissern, daß Ihr entsprechendes Assemblerprogramm fehlerfrei läuft.

Block-Read (B-R) und User 1 (U1)

Doch nun ran an den Speck: Öffnen Sie zunächst einen Direktzugriffskanal und richten Sie einen Floppypuffer ein:

```
SETLFS = $FFBA
SETNAM = $FFBD
OPEN = $FFC0
LDA #$02 ;Filenummer
LDX #$08 ;Geräteadress
LDY #$62 ;Sekundäradresse
JSR SETLFS
LDA #$01 ;Puffersymbol (=Raute,CHR$(35)
LDX #<name ;senden
LDY #>name
JSR SETNAM
JSR OPEN
...
name .text "#"
```

Das Rautenzeichen veranlaßt die Floppy, uns einen freien Puffer zur Verfügung zu stellen. Bei nachfolgenden Befehlen reicht es, nur die Sekundäradresse anzugeben: Schon wird der gewählte Puffer angesprochen! Welcher Bereich ausgewählt wurde, kann uns egal sein. Möchten Sie dennoch selbst darüber entscheiden, müssen Sie hinter der Raute die gewünschte Pufferzahl eintragen.

Die Speicherverteilung der Floppy-Datenpuffer:

- **0: $0300-$03FF,**
- **1: $0400-$04FF,**
- **2: $0500-$05FF,**
- **3: $0600-$06FF (reserviert für Directory),**
- **4: $0700-$07FF (BAM).**

Jetzt wollen wir einen Diskettensektor ins Floppy-RAM holen. Dazu brauchen wir den Befehlskanal:

```
...
LDA #$0F ;Filenummer
LDX #$08 ;Geräteadresse
LDY #$6F ;Sekundäradresse
JSR SETLFS
JSR OPEN
```

Wir haben uns für den ersten Directory-Block (Track 18, Sektor 0) entschieden. Geladen wird der Datenblock mit dem »Block-Read«-Befehl (B-R). Da der aber einen systembedingten Fehler hat, weichen wir auf die Abkürzung »User 1« (U1) aus, die mit B-R identisch ist:

```
CKOUT = $FFC9
STROUT = $AB1E
CLRCHN = $FFCC
LDX #$0F ;Floppy als aktuelles
JSR CKOUT ;Ausgabegerät setzen
LDA #<read ;ein String wird an
LDY #>read ;das aktuelle
JSR STROUT ;Ausgabegerät gesandt
JSR CLRCHN ;Ein-Ausgabe-(I/O)-Standard setzen
...
read .text "u1 2 0 18 0"
 .byte 0
```

Haben Sie's gemerkt? Jetzt unterhalten wir uns mit der Floppy in einer anderen Sprache: über die Befehle LISTEN oder TALK. Man setzt das Laufwerk als aktuelles Ausgabegerät und sendet die Daten über die Systembefehle. Dazu muß man nicht auf die langsame Ausgaberoutine $FFD2 (Byte für Byte) zurückgreifen, sondern kann die für ganze Zeichenketten (maximal 256 Byte) ab $AB1E benutzen (beliebigen String ans aktuelle Ausgabegerät senden). Als Kennzeichnung fürs Textende muß dort allerdings eine »0« stehen. Durch Übergabe der Dateinummer an die Routine CKOUT teilt man der Floppystation mit, daß sie Daten empfangen soll.

Die Sequenz des Block-Read-Befehls: Zunächst wird die Sekundäradresse des Files angegeben, über das ein Puffer reserviert wurde (also 2). Die Floppy sucht sich selbständig das zugeordnete RAM. Das nächste Datum ist die Laufwerksnummer: bei der 1541 stets »0« (nicht zu verwechseln mit der Geräteadresse!). Abschließend folgt der gewünschte Track und Sektor, der in den Puffer geholt werden soll.

Block-Pointer (B-P)

Jetzt haben wir die den Directory-Vorspann (BAM) im Floppyspeicher und wollen wissen, wie der Diskettenname lautet. Der beginnt im Sektor beim 144sten Byte. Bevor man den Namen auslesen kann, muß man einen internen Floppyzeiger auf die Position richten. Erst dann lassen sich die Werte über den Direktzugriffskanal holen:

```
LDX #$0F ;der Block-Pointer-
JSR CKOUT ;Befehl (B-P) wird an
LDA #<point ;die Floppy
LDY #>point ;übertragen
JSR STROUT
JSR CLRCHN ;Standard-I/O setzen
LDX #$02 ;Floppy als
JSR CHKIN ;Eingabegerät setzen
LDA #$0D ;Cursor in die
JSR BSOUT ;nächste Zeile
LDX #$0F
label1: JSR BASIN ;16 Zeichen auslesen
JSR BSOUT ;und auf den
DEX ;Bildschirm
BPL label1 ;ausgeben
JSR CLRCHN ;Standard-I/O setzen
point .text "b-p 2 144"
 .byte 0
```

Der Block-Pointer (B-P) braucht neben der Pufferinformation (2) noch die Zeigerposition (144). Dann wird das Laufwerk als Eingabegerät definiert, die Daten holt man in einer Schleife aus dem Puffer und gibt sie auf dem Bildschirm aus.

Block-Write (B-W) – User 2 (U2)

Möchten Sie den Diskettennamen ändern? Kein Problem: Block-Pointer erneut setzen, anschließend die geänderten Bytes über den Direktzugriffskanal senden und den gesamten Blockinhalt mit »Block-Write« (U2) wieder auf die Disk zurückspeichern!

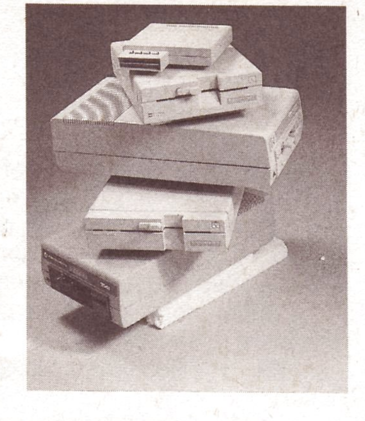
```
; *** zuerst Pointer setzen !!! ***
 ...
LDX #$02
JSR CKOUT
LDX #$00
label2: LDA neu,X ;neuen Disknamen
JSR BSOUT ;an Floppy
INX ;senden
CPX #$16
BNE label2
JSR CLRCHN
LDX #$0F ;Block-Write-Befehl
JSR CKOUT ;an Floppy
LDA #<write ;senden
LDY #>write
JSR STROUT
JSR CLRCHN
neu .text "*** TESTDISK ***"
write .text "u2 2 0 18 0"
 .byte 0
```

Der »Block-Write«-Befehl (B-W) wird ebenfalls durch die User-Anweisung U2 ersetzt: Man verwendet die gleiche Syntax wie beim Lesen (U1). Doch wenn Sie jetzt wieder den Disknamen in den Speicher holen, hat sich offensichtlich nichts geändert. Woran liegt's? Ganz einfach: Das Floppy-DOS erkennt Diskettenwechsel an der ID-Kennung, nicht am Namen der Disk! Das ist der Grund, warum man beim Formatieren nie die gleiche ID benutzen sollte - sonst macht man's Floppyfehlern allzu leicht!

Block-Allocate (B-A)

Es gibt noch drei weitere Blockbefehle, die man zur Floppy-Programmierung unbedingt kennen sollte:

»Block-Allocate« (B-A) kennzeichnet gewünschte Sektoren als »belegt«. Auf jeder Diskette existiert ein Verzeichnis, das



exakt Auskunft über bereits beschriebene Blöcke gibt – den Blockbelegungsplan (BAM, Spur 18, Sektor 0). Möchten Sie bestimmte Sektoren vor dem Überschreiben schützen, müssen Sie den Befehlskanal öffnen und den entsprechenden Befehl senden:

```
...
allocate .text "B-A 0 TR SE"
```

TR bedeutet die gewünschte Spur, SE ist der Sektor. Selbstverständlich muß man in die Befehlssequenz echte Zahlen statt Buchstaben eintragen (gilt auch für die folgenden Beispiele!). Außerdem läßt sich feststellen, ob ein Block bereits belegt oder noch frei ist. Man erfährt das, wenn man nach der Belegungsanforderung den Fehlerkanal liest. War der Sektor bereits reserviert, erhalten Sie die Meldung »No Block«. Zusätzlicher Service des Floppy-Betriebssystems: Hinter der Textmeldung erscheint die nächste freie Sektornummer.

Block-Free (B-F)

Natürlich kann man belegte Blöcke wieder freistellen: per »Block-Free«-Anweisung (B-F). Die Syntax ist nahezu identisch:

```
...
free .text "B-F 0 TR SE"
```

Block-Execute (B-E)

Der letzte Block-Befehl unserer Liste wird selten benutzt: Block-Execute (B-E). Er dient zum Übertragen eines 256 Byte großen Bereichs (z.B. ein Assembler-Programm) in einen vorher geöffneten Puffer mit anschließendem Start. Damit läßt sich ein Mini-Programm in Maschinensprache auf Diskette zur Floppystation schaufeln und starten:

Zuerst öffnet man wie gewohnt Kommando- und Direktzugriffskanal, dann überträgt man die Befehlssequenz für Block-Execute:

```
...
execute .text "B-E 2 0 TR SE"
```

Das maximal 256 Byte große Programm in Spur TR, Sektor SE, wird in den Puffer gelesen und dort gestartet.

Memory-Befehle und Floppy-RAM

Direkte Manipulationen innerhalb des Floppyspeichers mit 2 KByte RAM lassen sich per Memory-Anweisungen realisieren. Man kann Daten ins Floppy-RAM schreiben, diese lesen oder mit dem Betriebssystem arbeiten. Jede Kommunikation wird über den Befehlskanal abgewickelt.

Memory-Read (M-R)

Kommen wir gleich zur meist verwendeten Anweisung: Floppyspeicher lesen. In unserem Listingbeispiel wird ab Adresse $0790 der Diskettenname gelesen und zum C64 gesandt, der die Bytes an den Bildschirm weiterleitet.

```
SETLFS = $FFBA
SETNAM = $FFBD
OPEN = $FFC0
CHKIN = $FFC6
BASIN = $FFCF
BSOUT = $FFD2
CLRCHN = $FFCC
CLOSE = $FFC3
LDA #$0F ;Filenummer
LDX #$08 ;Geräteadresse
LDY #$6F ;Sekundäradresse
JSR SETLFS
LDA #$06 ;Befehlssequenz
LDX #<read ;senden
LDY #>read
JSR SETNAM
JSR OPEN
LDX #$0F
JSR CHKIN
LDA #$00 ;Statusregister
STA $90 ;löschen
label1: JSR BASIN ;Daten empfangen
JSR BSOUT ;und ausgeben
BIT $90 ;Bit 7 gelöscht?
BVC label1 ;dann Zeichen holen
JSR CLRCHN
LDA #$0F
JSR CLOSE
RTS
read .text "M-R"
 .byte $90,$07,$17
```

Beim Öffnen des Befehlskanals wird die »Memory-Read«-Anweisung initialsiert. Nachdem man die Eingabedatei zur Floppy umgeleitet hat, nimmt das Programm die angeforderten Daten entgegen. Nach der Kennung »M-R« muß man Low- und High-Byte der Startadresse angeben, das dritte Byte fixiert die gewünschte Anzahl der zu lesenden Bytes. Ist Bit #6 im Statusregister $90 gesetzt, sind die Daten korrekt rübergekommen.

Memory-Write (M-W) – Memory-Execute (M-E)

Mit diese beiden Anweisungen (in Personalunion) kann man Daten ins Floppy-RAM schreiben und eigene Assembler-Programme bzw. Betriebssystemroutinen aufrufen.

Unser Beispiel überträgt den Assembler-Befehl »JMP $C12C« in den Floppy-Pufferspeicher ab $0600 und aktiviert ihn. Zur Information: Ab $C12C steht die Floppy-ROM-Routine, die das Blinken der Leuchtanzeige nach einem Fehler einschaltet:

```
SETLFS = $FFBA
SETNAM = $FFBD
OPEN = $FFC0
CKOUT = $FFC9
STROUT = $AB1E
CLRCHN = $FFCC
CLOSE = $FFC3
LDA #$0F ;Filenummer
LDX #$08 ;Geräteadresse
LDY #$6F ;Sekundäradresse
JSR SETLFS
LDA #$09 ;Befehlsequenz M-W
LDX #<write ;übertragen
LDY #>write
JSR SETNAM
JSR OPEN
LDX #$0F
JSR CKOUT
LDA #<exec ;zweiten Befehl M-E
LDY #>exec ;übertragen
JSR STROUT
JSR CLRCHN
LDA #$0F
JSR CLOSE
RTS
write .text "m-w"
 .byte $00,$06,$03,$4c,$2c,$c1
exec .text "m-e"
 .byte $00,$06,$00
```

Der Memory-Write-Befehl ist im Aufbau identisch mit Memory-Read. Dann folgen die Daten-Bytes (abzulegen ab Adresse $0600):

- $00 $06: Low-High-Byte von Adresse $0600,
- $03: 3 Byte folgen...,
- $4C $2C $C1: Assembler-Code JMP $C12C.

Die Anweisung »Memory-Execute« braucht nur noch die Startadresse $0600. Das Byte $00 schließt die Befehlsfolge ab.

Zum Abschluß unseres Floppy-Programmierkurses steigen wir in die Praxis ein und entwickeln gemeinsam eine kurze Routine, die den Floppyfehlerkanal liest und den Text der Systemmeldung auf dem Bildschirm ausgibt. Versuchen Sie nun selbst, die Anweisungen zum Öffnen des Befehlskanals (ohne Dateinamen) per Assembler- oder Monitorprogramm einzugeben! Dann geht's weiter mit:

```
LDA #$00 ;Statusregister
STA $90 ;löschen
LDX #$0F ;Floppy als Eingabe-
JSR CHKIN ;gerät setzen
label1: JSR BASIN ;Daten empfangen
JSR BSOUT ;und ausgeben
BIT $90 ;Bit 6 gesetzt?
BVS label1
JSR CLRCHN ;Standard-I/O aktivieren
LDA #$0F ;File
JSR CLOSE ;schließen
RTS
```

Sie sehen, es ist auch für Assembler-Einsteiger gar nicht so schwer, eigene Floppyprogramme in Maschinensprache zu realisieren – vor allem, wenn man bedenkt, wieviel Arbeit uns die im C-64-Betriebssystem integrierten Routinen abnehmen! (Jörg Brokamp/bl)

Es muß nicht immer Assembler sein...

Wenn man sich die vorgestellten Maschinensprache-Routinen zur Floppyprogrammierung genauer ansieht, stellt man schnell fest, daß außer Öffnen und Schließen der Kanäle alle anderen Floppy-Befehle (z.B. Dateiname, B-R, M-W usw.) im Klartext übermittelt werden – also exakt wie Basic-Anweisungen!

Sicher sind Sie auch der Meinung, daß wir uns die Beschreibung zum Erzeugen sequentieller Dateien (SEQ, PRG, USR) mit der Floppy sparen können. Diese Art der Dateiverwaltung ist jedem geläufig (das zeigen die vielen Adreß- und Videoverwaltungen, die uns als Programmlistings erreichen) und gehören ins Einsteigerheft. Außerdem gibt's im Floppyhandbuch ausführliche Beispiele.

Komplizierter wird's bei den Direktzugriffsbefehlen (z.B. B-R, U1, M-W usw.), aber keine Panik: Die lassen sich in Basic ebenso problemlos aktivieren.

Um Bytes zu lesen, zu ändern und zurückzuspeichern, ist es auch hier nötig, den Floppybefehlskanal (Sekundäradresse 15) zu öffnen und einen Floppypuffer zur Datenaufnahme bereitzustellen. Das erledigt die Basic-Anweisung OPEN – inkl. Parameter- oder Befehlsübergabe:

```
OPEN 15,8,15: REM BEFEHLSKANAL
OPEN 1,8,2,"#": REM PUFFER RESERVIEREN
```

Möchte man einen bestimmten Puffer bereitstellen, muß man die Nummer angeben, z.B. #1 usw.

Denken Sie dran: Puffer 3 und 4 braucht die Floppy fürs Directory und BAM. Wenn Sie der Diskettenstation die Pufferwahl überlassen, gibt Ihnen das folgende Listing über die Nummer Bescheid:

```
100 OPEN 15,8,15
110 OPEN 1,8,2,"#"
120 GET#1, NR$
130 NR=ASC(NR$+CHR$(0))
140 CLOSE 1: CLOSE 15
150 PRINT NR
```

Die Block-Befehle (Zugriff auf Spuren und Sektoren) sind auch in Basic einfach zu realisieren. Die komplette Sequenz wird als String per PRINT# an die Floppy übermittelt. Dabei ist die Dateinummer für den Befehlskanal zu verwenden.

Hier ein Beispiel für **Block-Read (B-R):**

```
PRINT# fn, "B-R kn dn t s"
```

Die Kürzel sind innerhalb der Zeichenkette durch echte Zahlen zu ersetzen – oder Sie müssen eine Reihe CHR$-Codes aneinanderfügen:
- fn: Dateinummer (es gilt die des OPEN-Befehls für den Befehlskanal, also z.B. 15!),
- kn: Kanal (=Sekundäradresse beim OPEN-Befehl für den Puffer),
- dn: Laufwerksnummer (immer 0),
- t: Spur (Track),
- s: Sektor.

Will man z.B. den ersten Directory-Block (Spur 18, Sektor 1) von Diskette lesen und im Puffer ablegen, lautet die Basic-Anweisung:

```
PRINT#15,"B-R 2 0 18 1"
```

Die Angelegenheit hat einen Schönheitsfehler im System: B-R überliest das erste Byte eines Diskettensektors. Das gerade ist aber die Spurnummer des Folgeblocks einer Datei. Beim Zurückspeichern des Sektors können dann erhebliche Probleme auftreten, wenn die Zahl nicht bekannt ist. Profis weichen deshalb auf die User-1-Anweisung aus:

```
PRINT# 15, "U1 2 0 18 1"
```

Die 256 Byte eines Sektors lassen sich mit einer GET#-Schleife in den Computer holen.

Block-Write (B-W) schreibt Daten eines Pufferblocks wieder auf Diskette zurück – normalerweise in denselben, aber das hängt von Ihrer Befehlseingabe ab (es lassen sich beliebige andere Spur- und Sektornummern wählen):

```
PRINT#15, "B-W 2 0 18 1"
```

Obwohl die B-W-Anweisung korrekt arbeitet, gibt's dafür ebenfalls einen User-Befehl: U2 (ersetzt B-W)

Jeder Datenpuffer besitzt einen Zeiger **(Buffer-Pointer, B-P)**, der aufs aktuelle Byte weist und sich bei jedem Datenzugriff um »1« erhöht (damit lassen sich alle 256 Byte eines Datenblocks der Reihe nach lesen). Will man gezielt auf einzelne Werte oder Zeichenketten zurückgreifen, richtet man den Zeiger auf die entsprechende Position. Beispiel: das 122ste Byte in Spur 1, Sektor 16 interessiert Sie brennend! Verwenden Sie dazu dieses Demoprogramm:

```
100 OPEN 15,8,15
110 OPEN 1,8,2,"#"
120 PRINT#15, "U1 2 0 1 16"
130 PRINT#15, "B-P 2 122"
140 GET#1,A$
150 A=ASC(A$+CHR$(0))
160 CLOSE 1: CLOSE 15
170 PRINT A
```

Möchten Sie die ID-Kennung der aktuellen Disk im Laufwerk ändern? Dann laden Sie das Demoprogramm:

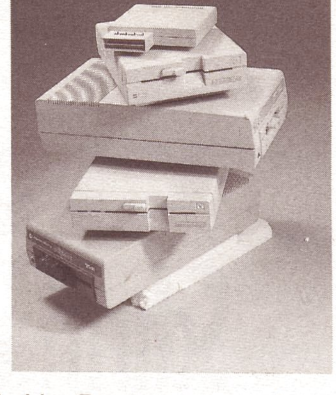
```
LOAD "5-BYTE-ID",8
```

und starten mit RUN.

Prompt erscheint die aktuelle ID auf dem Bildschirm (z.B. S2 2A), darunter die Frage: Neu?. Jetzt darf man bis zu fünf Zeichen eingeben, z.B. 64'er (mit <RETURN> bestätigen). Laden Sie nun erneut das Directory: Das Mini-Programm hat seine Schuldigkeit getan und die neue ID-Kennung eingetragen.

In Zeile 110 öffnet man Fehlerkanal und Datenpuffer. Zeile 120 weist die Floppy per Basic-Befehl PRINT# an, Sektor 0 der Spur 18 einzulesen (U1). In Zeile 130 wird der Blockzeiger aufs 162ste Byte positioniert (B-P): Dort stehen die fünf ID-Zeichen. Die Zeilen 140 und 150 lesen die Daten mit GET# und bringen sie auf den Bildschirm, ab Zeile 190 schreibt man den Pufferinhalt auf Disk zurück und schließt die offenen Datenkanäle.

Wer zur ID auch den Disknamen (den gesamten Disk-Header!) ändern will, muß das nächste Demoprogramm von der beiliegenden Sonderheftdiskette laden:



```
LOAD "NEW HEADER",8
```

Nach dem Start mit RUN erscheint der aktuelle Diskname auf dem Bildschirm. Geben Sie jetzt den neuen und die ID ein (ebenfalls bis maximal fünf Zeichen!): Der neue Header wird auf Diskette verewigt.

Wenn Sie sich die Listings beider Programme ansehen, werden Sie feststellen, daß diese ausgiebig mit den Befehlen U1, U2 und B-P jonglieren.

Schreibt man Daten mit Direktzugriffsbefehlen in bestimmte Disksektoren, muß man ihre Position in der BAM registrieren lassen. Gehören die Blocks nämlich zu keiner im Directory vermerkten Datei, könnten sie beim nächsten Speichern gnadenlos überschrieben werden. Um dies zu verhindern, gibt's die Anweisung **Block-Allocate (B-A)**. Unser Beispiel kennzeichnet Block 16 in Spur 1 als belegt:

Belegte Sektoren kennzeichnen

```
PRINT#15,"B-A 0 1 16"
```

Sie haben's sicher bemerkt: Die Kanalnummer 2 entfällt! War der Sektor schon reserviert, erhält man die Fehlermeldung:

```
65, NO BLOCK, XX, YY
```

XX und YY sind zweistellige Zahlen, die den nächsten freien Block bekanntgeben.

Die **Block-Free-Anweisung (B-F)** ist im Befehlssequenz-Aufbau identisch mit B-A und stellt der Floppy als belegt markierte Sektoren wieder zur Verfügung.

Der **Block-Execute-Befehl (B-E)** ähnelt der B-R-Anweisung. Zusätzliche Vorschrift: Die in den Datenpuffer gelesene Datei muß ein ausführbares Maschinenprogramm sein, das sofort loslegt.

Auf unserer Sonderheftdiskette finden Sie weitere Beispielprogramme zum Thema Block-Befehle, die man nahezu unverändert in eigene Basic-Programme einbauen kann. Beachten Sie dazu die REM-Erläuterungen in den Programmlistings. Noch ein Hinweis: Wenn Sie die Demoprogramme ausprobieren möchten, sollten Sie das nur mit Sicherheitskopien machen - nie mit Originalen!

Demoprogramm auf Diskette

- **Read Dir:** ein Unterprogramm, das einen Dateieintrag aus dem Directory liest. Da es in Zeile 1180 mit RETURN abschließt, darf man es nicht mit RUN starten! So kann man das Programm nach dem Laden ausprobieren (Direkteingabe in einer Bildschirmzeile):

```
CLR: GOSUB 1000: PRINT DD$
```

- **Write Dir:** das zweite Unterprogramm, um geänderte Einträge ins Directory zurückzuschreiben (wird mit GOSUB 2000 aufgerufen!).

Beide Unterprogramme unterstützen die Funktionen der beiden nächsten Beispielprogramme:

- **No Scratch:** schützt Dateien auf Diskette gegen Löschen. Als Markierung erscheint hinter dem Namen eine Spitzklammer »<«,
- **Dir-Sorter:** sortiert jedes Disketteninhaltverzeichnis alphabetisch und überträgt die neue Reihenfolge ins Directory. Bei umfangreichen Inhaltsverzeichnissen kann das allerdings einige Minuten dauern.
- **Disk-Protect:** Statt einzelner Files wird die komplette Diskettenseite gegen Löschen oder Überschreiben geschützt: Nur Formatieren mit ID-Kennung umgeht diese Schutzvorrichtung! Soft-Formatieren oder VALIDATE ist nicht möglich!

Mit der nächsten Gruppe der Floppy-Direktzugriffsbefehle geht's ans Eingemachte: Im Gegensatz zu den Block-Befehlen, die auf Spuren und Sektoren der externen Disk zugreifen, erlauben die Memory-Anweisungen, jede gewünschte interne Speicherstelle der Floppy zu lesen (also auch das ROM!) und zu beschreiben (nur Floppy-RAM) - vergleichbar mit POKE- und PEEK des Basic 2.0! Auch hier ist zunächst der Floppybefehlkanal zu öffnen, die Reservierung eines Datenpuffers entfällt. Die Befehlssequenz hat sich gegenüber den Block-Befehlen geändert: Statt Spur- oder Sektornummer muß nach einer Memory-Anweisung die gesuchte ROM- oder RAM-Adresse als Low-High-Byte stehen (dort wird der Zeiger positioniert), zum Abschluß gibt man an, wie viele Bytes man lesen (oder schreiben) will.

Ein Beispiel für **Memory-Read (M-R)**: Lesen beider ID-Zeichen der zuletzt initialisierten Diskette im ASCII-Format:

```
10 OPEN 15,8,15,"I"
20 PRINT#15, "M-R" CHR$(18)CHR$(0)CHR$(2)
30 GET#15, A$, B$
40 PRINT A$;B$
50 CLOSE 15
```

Damit positioniert man den Befehlszeiger auf Adresse 18 in der Floppy-Zeropage: dort werden beide ID-Zeichen gespeichert.

Memory-Write entspricht POKEs in den Floppy-RAM-Speicher. Nach Angabe der Adresse (Low-High-Byte) und Anzahl der zu übertragenden Bytes folgen beliebig viele Daten, als CHR$ deklariert. Folgende Sequenz ändert z.B. die ID in der Floppy-Zeropage (Adressen $12/$13) in »AB«:

Memory-Befehle in Basic

```
PRINT#15, "M-W" CHR$(18) CHR$(0) CHR$(2) CHR$(65)
CHR$(66)
```

Auch den SYS-Befehl des Basic 2.0 kann man mit Memory-Anweisungen simulieren: **Memory-Execute** springt zur angegebenen Adresse (Low-High-Byte-Format) und startet das dort gespeicherte Maschinenprogramm. Unser Beispiel im Assembler-Teil dieses Floppykurses (Blinken der LED wie bei einem Diskfehler) sieht in Basic so aus:

```
10 OPEN 15,8,15
15 REM SPRUNG ZU ADRESSE $C12C:
20 PRINT#15, "M-E"CHR$(44)CHR$(193)
30 CLOSE 15
```

Beachten Sie auch die REM-Zeilen unseres Demoprogramms »LED-Blinker« auf der Diskette zu diesem Sonderheft.

Das nächste Beispiel zu M-W, »Soft Floppy«, überträgt den Wert 15 zur Adresse $1C07 (7175) des Floppy-RAM. Nach RUN verhindert das Mini-Programm, daß der Schreib-Lesekopf beim harten Formatieren (mit ID) ans Gehäuse schlägt (verhindert das bekannte Rattern, das den Schreib-Lesekopf dejustiert). Außerdem beschleunigt es Ladeoperationen um den Faktor 3.

Wer inzwischen nach Memory-Befehlen süchtig ist: Die beiden Programme »Read-Error 22« und »Read-Error 23« verwenden die Anweisung M-E, um Maschinencode zu starten, der per Schleife und M-W im Floppypuffer 2 ab $0500 abgelegt wird.

Genießen Sie die beiden Demoprogramme mit Vorsicht: Sie erzeugen nach RUN die DOS-Fehler 22 (Datenblockkennung zerstört) bzw. Nr. 23 (Prüfsummenfehler) auf der aktuellen Diskette im Laufwerk. Überprüfen Sie's mit »Sir-Scan« auf dieser Sonderheftdiskette! Also nochmals die eindringliche Warnung: nie Originaldisketten verwenden (vor allem nicht die Diskette zu diesem Floppy-Sonderheft Nr. 83) - sonst könnte Ihnen die Freude über neue Erkenntnisse aus unserem Floppykurs schnell und gründlich vergehen! (bl)

Jetzt geht's erst richtig los bei Spur 36!

Schritt über die magische Grenze

Es gibt jede Menge kommerzielle Software, die bei 5¼-Zoll-Disketten auf höhere Spuren als Nr. 35 zugreift. Wir zeigen Ihnen, wie man eigene Floppyscheiben manipulieren kann und was man mit dem Spurenüberschuß anfängt.

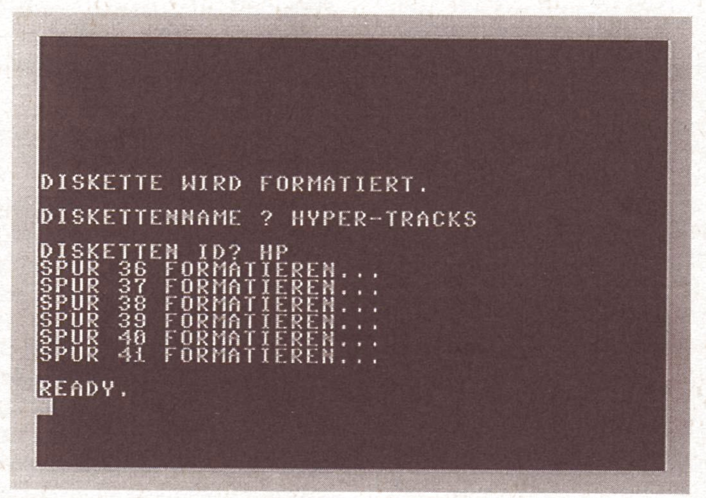
Das war vom Hersteller nicht beabsichtigt: Mit einem Trick kann man noch bis zu fünf, unter günstigen Hardwarevoraussetzungen auch sieben zusätzliche Datenspuren auf der Disk einrichten und mit Daten beschreiben!

Es gibt allerdings einen berechtigten Grund, warum das Betriebssystem der 1541 nur maximal 35 Spuren auf einer 5¼-Zoll-Disk formatiert: Vor allem, wenn die Justage des Floppykopfs nicht mehr stimmt, kann's bei höheren Spuren zu erheblichen Schreib- oder Lesefehlern kommen - statt wertvoller Daten bekommt man dann bloß noch Schrott!

Will man die magische Grenze »35« überschreiten, muß man das Floppy-DOS (Disk Operating System) überlisten: Ab Adresse $FAC7 (64199) im Floppy-ROM befindet sich eine Routine, die den logischen Ablauf beim Formatieren steuert. Man aktiviert dieses Maschinenprogramm durch einen Befehlscode für Datenpuffer 3 (ab $0600 im Floppyspeicher). Die Routine gibt das Steuer erst wieder aus der Hand, wenn die letztmögliche Spur (also Nr. 35) formatiert wurde.

Floppy-DOS formatiert nur 35 Spuren

Bevor das Betriebssystem mit dem Formatieren loslegt, müssen auch alle nötigen Angaben in der Zeropage des DOS vorhanden sein: In Adresse $51 steht die Startspur (normal »1«). Dieser Wert wird beim Formatieren ständig erhöht. Gleichzeitig existiert ein Zeiger auf einer Tabelle, in der die Sektorenanzahl pro aktueller Diskettenspur vermerkt ist (wird in Adresse $43 abgelegt). Die DOS-Formatierroutine speichert diesen Wert und reduziert ihn ständig. Erreicht er »0«, muß die Spurnummer in Speicherstelle $51 erhöht werden - so lange, bis Spur 35 erreicht ist. Diese ominöse Zahl (= letzte erreichbare Spur) ist als Konstante im ROM-Speicher des Laufwerks enthalten und läßt sich nicht ändern. Aber jetzt kommt's: Die Nummer der Anfangsspur, bei der die Formatierung beginnt, muß unbedingt im Floppy-RAM liegen (sonst könnte man ja den Wert nicht ständig anpassen und hochzählen!). Die Zahl erhöht sich automatisch, wenn die Sektorenkapazität der Spur erreicht ist - also kommt es auf die Nummer der Startspur an (man beginnt eben mit einem höheren Wert als »35«!).



[1] Jede Superspur von 36 bis 41 muß einzeln formatiert werden

[2] Pro Sektor lassen sich maximal 240 Textbytes eingeben, die in einer der Hyper-Spuren gespeichert werden

```
SEKTOR SCHREIBEN IN SPUR 36 BIS 40
Richtige Diskette einlegen, dann
sechs Bildschirmzeilen Text eingeben.
Jede Zeile mit <RETURN> abschliessen!

DER TRICK MIT DER ANFANGSSPUR

Um ein Assembler-Programm fuer die
Floppy zu starten gibt es zwei
Moeglichkeiten:
DATAs in Basic-Zeilen ablegen!
```

Jedoch: grau ist alle Theorie! Das 1541-DOS kennt keine Basic-Anweisung oder entsprechende Befehle, um die Formatierung bei einer beliebigen Spur beginnen zu lassen. Die übliche Befehlssequenz zum Formatieren einer Diskette:

`OPEN 1,8,15,"N:(Name),ID": CLOSE 1`

stellt automatisch Anfangsspur 1 ein, setzt die entsprechenden Parameter in der Floppy-Zeropage (z.B. die Zeiger auf die erwähnte Sektorentabelle). Außerdem legt die Formatierroutine im Vorspann ausdrücklich Track 1 als Startspur fest.

Um das DOS zu überlisten, muß man genau hier ansetzen: Eigene Maschinenprogramme im Laufwerks-RAM werden nämlich problemlos akzeptiert. Dazu stellt die Floppy einen Speicherbereich von 2 KByte zur Verfügung - man kann aber davon nur 1380 Byte Pufferspeicher nutzen (von Adresse $0300 bis $07FF). Den Bereich von $00 bis $02FF braucht das DOS als eigenen Arbeitsspeicher.

Um ein Assembler-Programm für die Floppy zu starten, gibt's zwei Möglichkeiten: Es ist als Datas in Basic-Zeilen abgelegt, die an den richtigen Speicherbereich gePOKEt werden oder wird als Maschinenprogramm nachgeladen. Wir haben uns für die zweite Form entschieden. Per OPEN-Anweisung in Zeile 10110 wird Puffer 2 zur Datenaufnahme geöffnet. Ab Zeile 10240 liest das Programm die Daten von »FP.Formater.obj« und legt sie dort ab. Das Maschinenprogramm ersetzt die Systemroutinen, die beim regulären NEW-Befehl (N:) aktiviert werden und umgeht damit die Tatsache, daß der Formatiervorgang grundsätzlich bei Spur 1 beginnt. Jetzt kann man theoretisch beliebige Nummern über »35« eintragen - aus hardwaretechnischen Gründen sind aber nur Werte von 36 bis maximal 41 sinnvoll.

Der Trick mit der Anfangsspur

Probieren Sie's aus und laden Sie das Programm mit:

```
LOAD "FORMAT 1-41",8
```

und starten Sie mit RUN. Nachdem die Floppy initialisiert und das Assembler-Programm »FP.Formater.obj« nachgeladen wurde, müssen Sie die aktuelle Diskette entfernen, eine andere einlegen (Achtung: Sie wird vollständig neu formatiert!) und <RETURN> drücken.

Zunächst richtet das Programm die gewohnten Spuren 1 bis 35 ein, dann tritt das Assembler-Programm in Aktion (es arbeitet im Floppypuffer $0300): Nacheinander werden die Spuren 36 bis 41 (Abb. 1) erzeugt und die Sektoren mit dem Byte-Wert $01 beschrieben.

Bevor das Formatieren beginnt, muß das Programm die nötigen Vorbereitungen treffen, z.B. die entsprechenden DOS-Parameter in der Floppy-Zeropage setzen. Dazu stehen die Floppyanweisungen USER 3 (U3) bis USER 8 (U8) zur Verfügung. In unserem Beispielprogramm übergibt z.B. der U3-Befehl die Anfangsspur. Beim normalen Formatiervorgang (Spuren 1 bis 35) steht der Sprungbefehl zur Betriebssystemroutine in Adresse $0600 (das ist der Anfang des Floppy-Datenpuffers 3). Bei unserem selbstgestrickten Assembler-Programm ist dort eine Steuerungsroutine abgelegt, die das Formatieren auf höhere Spuren als »35« einleitet. Um die DOS-Routine mit den USER-Anweisungen zu erreichen, muß man zuvor den entsprechenden Sprungbefehl und die Bytes der ID-Kennung im Datenpuffer 2 ablegen.

»Format 1 - 41« formatiert stets die gesamte Diskette! Was aber, wenn man auf vollbeschriebenen Datendisketten nur die oberen Spuren 36 bis 41 einrichten möchte (um z.B. ein Kennwort, den Namen des Anwenders oder sonstige Infos dort unterzubringen)? Dann müssen Sie unsere modifizierte Programmversion benutzen:

```
LOAD "FORMAT 36-41",8
```

Nach dem Start mit RUN sucht dieses Utility ebenfalls nach der Assembler-Datei »FP.Formater.obj«, die sich auf derselben Diskette befinden muß. Die normale Formatierroutine für die Spuren 1 bis 35 wurde aus der neuen Version entfernt und durch einen anderen Programmteil ersetzt: Einlesen der ID-Kennung der aktuellen Diskette im Laufwerk (Zeilen 10614 bis 10670). Diese beiden Bytes werden diesmal nämlich nicht von Ihnen eingegeben, da die Komplett-Formatierung entfällt. Dennoch muß das Programm unbedingt die Werte bekommen, um höhere Spuren korrekt zu formatieren! Dazu öffnet man den Fehler- (OPEN 15,8,15) und einen Datenkanal zum nächsten freien Puffer (OPEN 2,8,2). Per U1-Befehl liest man den markanten Sektor (Spur 18, Block 0 = BAM) in den Datenpufferspeicher. Die Feinstellung (Kopf auf die bewußten Bytes positionieren) erreicht man mit der »B-P«-Anweisung: Ab Speicherstelle 162 stehen in der BAM wenigsten zwei, maximal fünf signifikante Zeichen (ID-Kennung). Relevant fürs Programm sind nur die ersten beiden Bytes. Anschließend formatiert das Utility die Spuren 36 bis 41, läßt aber andere Daten in tieferen Spuren in Ruhe.

Kurzinfo: Trackwrite/Trackread

Programmart: Utilities zur Manipulationen der Spuren 36 bis 41
Laden: LOAD "TRACKWRITE",8 bzw. LOAD "TRACKREAD",8
Starten: nach dem Laden mit RUN
Besonderheiten: speichert Daten jeweils nur auf einen Sektor in den oberen Spuren. Kann in den Programmen in den Zeilen 10 bis 20 geändert werden.
Benötigte Blocks: 5
Programmautor: Michael Hoos

Kurzinfo: Format 1 – 41/Format 36 – 41

Programmart: Formatier-Utilities für Floppyspuren über »35«
Laden: LOAD "FORMAT 1-41",8 bzw. LOAD "FORMAT 36-41",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: Programmfunktionen sind zwar bis Spur 41 eingestellt, aber nur wenige Laufwerke schaffen es, diesen Track zu erzeugen!
Benötigte Blocks: 7
Programmautor: Michael Hoos

Daten für höhere Spuren lesen und schreiben

Doch was nützen zusätzlich gewonnene Tracks und Sektoren auf Diskette, wenn außer $01-Bytes nichts drin steht? Abhilfe schaffen weitere Hilfsprogramme zur Manipulation höherer Spuren.

Laden Sie zunächst:

```
LOAD "TRACKWRITE",8
```

Nach dem Start mit RUN bringt der Bildschirm einen blinkenden Cursor, hinter dem Sie Text bis zur Länge einer Bildschirmzeile (40 Zeichen) eingeben können (Abb. 2) - dann sollten Sie mit <RETURN> abschließen. Sie dürfen jedes Tastaturzeichen verwenden (Buchstaben, Zahlen, Grafikzeichen usw.). Von inversen Steuercodes (Cursor-Bewegungen, Tastenkombination mit <CTRL> usw.) sollte man aber die Finger lassen. Es folgen nochmals fünf Zeilen, dann ist Schluß: Die maximale Datensatzlänge darf 252 Bytes nicht übersteigen (Rechnen Sie's nach: 6 Zeilen x 40 Zeichen sind schon 240 Bytes!). Da die Eingaben für diese simplen Beispielprogramme vom INPUT-Befehl gesteuert werden, müssen Sie auch dessen Besonderheiten beachten: Komma, Semikolon und Doppelpunkt sind im Eingabetext tabu!

Text in oberen Spuren speichern

Vorher wird man aufgefordert, eine vorbereitete Diskette mit formatierten Spuren 36 bis 41 einzulegen. Der Eingabetext wird anschließend im Sektor 12 von Spur 37 abgelegt. Soll's woanders sein, müssen Sie die Werte für T und S in den Zeilen 10 bis 20 ändern.

Wer Daten auf den Bildschirm holen möchte, die in den oberen Spuren gespeichert sind, muß das zweite Utility laden:

```
LOAD "TRACKREAD",8
```

17 Blocks pro Hyperspur

Nach RUN wird man ebenfalls aufgefordert, eine hochspurige Diskette einzulegen und <RETURN> zu drücken. Nach kurzer Suche entdeckt das Laufwerk den gewünschten Sektor und bringt die Daten auf den Bildschirm. Aus Gründen der Kompatibilität greift das Programm ebenfalls auf den Sektor 12 der Spur 37 zu (ändern wie gehabt!).

Die Spuren 36 bis 40 enthalten jeweils 17 Sektoren (von 0 bis 16) 256 Bytes, das entspricht einem zusätzlichen Speichervolumen von 85 Blocks oder 21 760 Bytes, die man aber in der Normalkonfiguration des Laufwerks weder zum Speichern von Basic-Programmen noch zur Datenablage z.B. von Adreßverwaltungsprogrammen verwenden kann.

Schnell stellt man bei solchen Hyperspur-Manipulationen fest, daß manche Floppystationen die Spur 40 gerade noch so eben schaffen. Unsere Testfloppies 1541-II und das 1571-Laufwerk des C128D waren z.B. nicht zu überreden, Track 41 zu bearbeiten. Auch klappte das Lesen der Daten nicht immer einwandfrei, wenn die Disketten von unterschiedlichen Laufwerken formatiert oder beschrieben wurden. Hier kommt's auf exakte Justierung und Einstellung des Schreib-Lesekopfs an. Nostalgische Erinnerungen an Zeiten werden wach, in denen man noch mit der Datasette hantierte! Wer das Floppy-DOS austricksen will, muß eben kleine Mängel kommentarlos akzeptieren. (bl)

Ultra Boot Menue – High-Speed-Disklader

Für Individualisten

Programme per Tastendruck blitzschnell von Diskette laden – der Wunschtraum jedes C-64-Freaks! »Ultra Boot Menue« macht's möglich: Stellen Sie sich für jede Disk Ihrer Sammlung ein spezielles Lademenü zusammen!

Die Vorzüge unseres Diskladers auf jeder Sonderheftdiskette sind nicht zu übersehen: Auf Knopfdruck aktiviert man alle registrierten Programme – ohne umständliche Ladeanweisungen. Der Nachteil: Er gilt nur für die jeweils aktuelle Disk unserer Sonderhefte und läßt sich ohne komplizierte Anpassung nicht für beliebig andere Disketten Ihrer Privatsammlung anwenden.

Ultra Boot Menue dagegen ist universell: Mehrere Dateien werden zu einem File-Pool zusammengeschlossen, die man nacheinander per Fastloader in den Computer holen und sogar automatisch starten kann!

Das Menüpaket besteht aus mehreren Programmteilen. Laden Sie zunächst den wichtigsten:

```
LOAD "ULTRAMENUE MAKER",8
```

Nach dem Start mit RUN werden Sie nach den Namen gefragt, die Sie im Lademenü verankern möchten. Legen Sie nun die gewünschte Disk ins Laufwerk (<$> bringt das Directory!) und tragen Sie die Programmnamen korrekt ein. <RETURN> im nächsten leeren Fragefeld beendet die Eingabe. Der Bildschirm bringt die gewählten Namen nochmals. Sind Sie mit der Auswahl einverstanden, geht's per <J> weiter.

Jetzt gibt man bekannt, wie die Programme zu starten (Abb. 1) sind:

- <S> für SYS-Befehl (dann muß man zusätzlich die Startadresse angeben),
- <R> mit RUN oder
- <T> als Tool-Datei (s. Beschreibung zu »Ultra Tool«).

```
*** Geben Sie jetzt ein, wie die Files
    gestartet werden .
      s : Sys
      r : Run
      t : Tool-File

File 1  : c64-tools
==> Startbefehl (s/r/t) ? r
File 2  : double-format
==> Startbefehl (s/r/t) ? r
File 3  : track-copy ii
==> Startbefehl (s/r/t) ? r
File 4  : supercopy
==> Startbefehl (s/r/t) ? r
```

[1] Wichtig: Korrekte Startanweisung des Programms angeben!

Die nächste Abfrage legt die Reihenfolge für die Menüdatei fest: Geben Sie die gewünschten Nummern ein (z.B. 1, 4, 5 usw.). Auch hier endet die Abfrage ohne Zahleneingabe per <RETURN>. Das Programm sieht noch einen zweiten Datensatz vor, der aber nur bei etlichen gewählten Dateinamen zum Tragen kommt: sonst auch hier nur <RETURN> drücken!

Anschließend kann man festlegen, ob Programme nach dem Laden per Ultra Menü automatisch starten sollen.

Das folgende Menü bietet zwei Möglichkeiten:

- <0>: Damit stellt man einen File-Pool zusammen,
- <1>: erzeugt das Lademenü mit den auf dem Screen gezeigten Dateien.

Die Programme werden nach <RETURN> und Tastendruck geladen, bearbeitet (z.B. für den Autostart) und das Ultraboot-Menü generiert. Zum Sichern dieser Datei auf Diskette wählen Sie einen markanten Dateinamen (z.B. MENU1.SRC). Per Taste <J> kann man eine weitere Diskette bearbeiten, mit <N> verläßt man Ultra Menue Maker.

Boot-Datei erzeugen

Jetzt befindet sich auf der Disk zwar die Menüdatei – aber sie läßt sich noch nicht einsetzen! Der nächste Arbeitsgang steht bevor. Menu1.Src ist quasi eine Quelldatei (Source-Code, daher die vorgeschlagene Endung .SRC!), die sich nur mit einem speziellen Ladeprogramm korrekt starten läßt.

Die Vorbereitungen dazu erledigt das nächste Utility auf unserer Sonderheftdiskette:

```
LOAD "ULTRABOOT MAKER+",8
```

Wenn nach RUN der Arbeitsbildschirm erscheint, trägt man bei der Frage »Programmname?« die Menü-Quelldatei ein (also z.B. Menu1.src) ein. Beachten den wichtigen Hinweis in der dritten Bildschirmzeile: 104 Blöcke maximale Dateilänge! Sollte Ihre SRC-Datei größer sein, müssen Sie Ultra Menue Maker erneut starten, weniger Files pro Diskmenü wählen und mehr, aber kürzere Quelldateien erzeugen.

Ultraboot Maker lädt den Source-Code, gibt dessen Block- und Byte-Anzahl aus und generiert im Computer das endgültige Ladeprogramm (Abb. 2). Nach Tastendruck wird es auf die aktuelle Diskette im Laufwerk gespeichert – und verbraucht dennoch keinen wertvollen Speicherplatz auf Disk. Abgesehen von den Hyper-Spuren 36 bis 40: Diese Tracks werden nun zunächst formatiert, die Quelldatei in den Computer geholt, in den oberen Spuren abgelegt (der Bildschirm zeigt, wo!) und der Lader in einem freien Sektor der Spur 18 abgelegt – deshalb belegt er keinen Block auf den Spuren 1 bis 17 und 19 bis 35.

[2] Das Ultramenü wird in den oberen Tracks 36 bis 40 abgelegt

```
 *** Ultraboot Maker ***
* (c) 1992 Martin Pfost
 *** maximale Filelaenge : 104 Bloecke

Ultraboot Maker erstellt auf Diskette
ein U-File, was sehr schnell ein be -
stimmtes Programm laedt .

 *** Programm laden
==> Programmname = ? t1
==> Diskette mit 't1' einlegen !
  't1' wird gelesen .
    Bloecke :  15
    Bytes   : 3650
 *** Ultraboot Menue-File !
==> Diskette, die Ultraboot erhalten
  soll, einlegen !
```

Das Ladeprogramm besteht nur aus einem Buchstaben: U (der Name der Quelldatei wird lediglich der Übersicht wegen angefügt, falls man mehrere Boot-Files in einem Durchgang erzeugt).

Ab sofort brauchen Sie den Quell-Code (z.B. Menu1.src) nicht mehr und können ihn SCRATCHen: Das wichtigste Programm auf Diskette heißt jetzt »U«!

Wollen Sie's ausprobieren? Dann laden Sie das Boot-Programm:

```
LOAD "U",8,1
```

Das Programm startet automatisch. Nach kurzem Zugriff auf die oberen Spuren bringt der Bildschirm das Boot-Menü mit den beiden Auswahlmöglichkeiten (Abb. 3):

- <0>: Programme per entsprechender Zahlentaste wählen,
- <1>: ruft das rechts daneben vermerkte Programm auf. Ist es revers gekennzeichnet, startet es sofort nach dem Laden.

Wer die »U«-Datei mit den diversen Directory-Sortierfunktionen an erster Stelle im Diskinhaltsverzeichnis plaziert, kann das individuelle Diskettenmenü künftig mit dem Joker <*> laden.

Zuviel des Guten...

Die Funktionen des Ultramenüs klappen nicht mit Dateien, die den Bereich von $0210 bis $03FF verändern: Exakt dort werden die Laderroutinen der Boot-Datei U abgelegt!

Außerdem ist U (glücklicherweise nur in seltenen Fällen!) zu übereifrig: Es lädt auch den letzten Block eines Programms vollständig von Diskette, obwohl es dort eventuell nur noch wenige Bytes braucht, um mit der exakten Endadresse abzuschließen! Der Grund: Das Floppy-Betriebssystem teilt

```
*** Ultraboot Menue ***     Martin Pfost
     (c) 1986 Heureka Teachware

0 Files aussuchen
1 dual-filecopy

Ihre Wahl : ■
```

[3] Bequemer geht's nicht: gewünschte Taste drücken – Programm lädt und startet automatisch!

Kurzinfo: Ultra Menue Tools/Ultraload Plus

Programmart: Disk-Tool
Laden: LOAD "ULTRAMENUE MAKER",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: Disketten-Lademenü wird in den oberen Spuren (36 bis 40) abgelegt und Boot-Datei in Spur 18 erzeugt. Speeder »Ultralaod Plus« läßt sich auch separat aktivieren.
Benötigte Blocks: 78
Programmautor: Martin Pfost

die Programmlänge nicht in Sektoren mit 256 Byte, sondern kann nur 254 berücksichtigen. Die ersten beiden Bytes sind als Verbindungszeiger (Spur, Sektor) zum logisch nächsten Block reserviert. Dazu ein Beispiel:

Ein Maschinensprache-Monitor, der mit »,8,1« zu laden ist, beginnt z.B. bei Adresse $C000 (49152) und endet mit dem letzten Byte in Speicherzelle $CFFF (53247): Das sind 4096 Byte. Geteilt durch »256« erhält man exakt 16 Sektoren auf Disk, aber 4096 : 254 ergibt 16,126 (also 16 Blocks 254 Byte und 32 Byte im 17ten!). Da das Floppy-DOS keine Achtel- oder Viertelsektoren kennt, wird immer der gesamte Block als belegt gekennzeichnet. Wenn man nun das Assembler-Programm mit Ultramenü bearbeitet, werden künftig alle 17 Blocks **vollständig** geladen – also auch die restlichen 222 Byte vom 17ten Sektor, die mit dem Monitorprogramm absolut nichts zu tun haben. Schlimmer noch: Durch die spezielle Speicherposition, in unserem Beispiel bewußt so gewählt, würden die Rest-Bytes beim Laden den I/O-Bereich des C64 überschreiben: vorprogrammierter Computerabsturz!

Jetzt kommt unser Heinzelmännchen zum Zug: »Ultramenü Tool« macht aus absolut zu ladenden Assembler-Programmen solche mit Basic-Start: Dem Basic-RAM ist es egal, ob z.B. 222 Byte mehr oder weniger geladen werden. Nach RUN wird die Assembler-Datei in den vorgesehenen Bereich ab $C000 gePOKEt und hört exakt bei $CFFF auf! Ab sofort: null problemo!

Laden Sie das Utility mit:

```
LOAD "ULTRAMENU TOOL",8
```

Nach dem Start mit RUN fragt das Programm nach der Assembler-Datei, die man bearbeiten will (Abb. 4). Die Startadresse (entsprechender SYS-Befehl!) sollte man unbedingt kennen: Sie wird als nächstes eingetragen (dezimal, in unserem Beispiel also 49152). Dann erzeugt das Programm auf Diskette ein Assembler-Programm mit Basic-Start: Es läßt sich mit RUN starten. Beim LIST-Befehl erscheint lediglich die Basic-Zeile:

```
1986 SYS 2062
```

Der neue Programmname beginnt zur Unterscheidung mit einem Punkt. Will man so konvertierte Dateien mit Ultra Menue Maker in die Boot-Datei U einbinden, markiert man den Startbefehl per Taste <T> (Tool-Datei).

```
100 rem *** ultramenue tool
110 rem *** 1992 martin pfost
120 rem *** (c) 1992 heureka teachware
ready.
run
==> Filename ? dual-filecopy
==> Startadresse ? 2049
==> entsprechende Diskette einlegen !
 Anfangsadresse des Files : 51143
ready.
```

[4] Ultra Menue Tool: macht aus jedem Assembler-Programm eins mit Basic-Start!

Schnelle Laderoutine inbegriffen

Beim UltraBoot Maker+ ist er bereits integriert und wird beim Erzeugen der Boot-Datei U automatisch eingebunden: »Ultraload Plus«, der Software-Lade-Speeder mit dem Faktor 6!

Falls Sie trotz aller Vorteile kein individuelles Lademenü auf Ihren Disketten einrichten möchten: Auf den Schnellader müssen Sie dennoch nicht verzichten.

Bevor Sie z.B. ein großes Spiel, eine umfangreiche Textverarbeitung oder Dateiverwaltung laden, sollten Sie das Utility aktivieren:

```
LOAD "ULTRALOAD PLUS",8
```

und mit RUN starten – der Schnellader bleibt eingeschaltet, bis man <RUN/STOP RESTORE> oder den Reset-Knopf drückt. Dann muß man ihn per SYS 336 erneut aktivieren.

Das überzeugt jeden Zweifler: Bei unserem Test verbrachte das normale Floppy-DOS zwei Minuten damit, ein 118 Blocks langes Basic-Programm zu laden – mit Ultraload Plus war's in 15 s vorbei! Und das Wichtigste: Der Floppy-Speeder arbeitete problemlos mit allen Testfloppystationen zusammen: 1541 und 1541-II, 1571-Single-Drive und internes Laufwerk des C 128D. Nur die 1581 mit der DOS-Version 10.0 ignoriert den Schnellader – sie lädt nämlich schon in der Normalkonfiguration schneller als die 1541! (bl)

Disc-Scanner 40 – spürt illegale Tracks auf

Disketten aufbohren

Vergessen Sie alle Diskmonitore, die Sie bisher verwendet haben: »Disc-Scanner 40« enthüllt auch Daten in unzugänglichen Bereichen: den Spuren 36 bis 40!

Diskettenmonitore sind des Profis liebstes Werkzeug, um damit bestimmte Bytes oder Inhalte ganzer Sektoren auf der Diskette aufzuspüren, sie zu ändern und wieder zurückzuschreiben. Die meisten Diskmonitore akzeptieren die üblichen Spuren von 1 bis 35, unser Programm erreicht aber auch höher gelegene Regionen ab Spur 36 bis maximal 40. Damit kann man Informationen in diesen Hyper-Spuren verewigen, lesen oder editieren. Vor allem kommerzielle Programme brauchen und nutzen die Möglichkeit, die Speicherkapazität einer Diskette durch zusätzliches Installieren der Spuren 36 bis 40 zu erhöhen, aber auch, um Kopierschutzdaten dort abzulegen – für Basic-Programmierer ist dieses Verfahren allerdings mit einem Wust an Direktzugriffsbefehlen und exakten Parametern verbunden. »Disc-Scanner 40« macht's quasi im Vorübergehen!

Laden Sie das Programm mit:

```
LOAD "DISC-SCANNER 40",8
```

und starten Sie mit RUN.

Der Menübildschirm (Abb. 1) ist in vier Felder gegliedert:
- Auswahlmenü: Die Funktionen ruft man per zugehöriger Zifferntaste auf. Befolgen Sie die anschließend erscheinenden Meldungen auf dem Bildschirm!
- Statuszeile oben: ...zeigt den Fehlerkanal. Wartet das Programm auf einen Funktionsaufruf, steht dort »Please choose«.
- Ein- und Ausgabefeld: ...bringt den Inhalt eines gewählten Diskettensektors zum Betrachten oder Editieren auf den Bildschirm.
- Untere Meldezeile: ...zeigt die Nummer der aktuellen Spur oder des Sektors, den Sie gerade bearbeiten bzw. gibt Programmaufforderungen aus (z.B. »Please switch on disc drive« = Floppy einschalten!).

```
DISC-SCANNER 40 (C) 1992 MARC HAWLITZECK
 (1) READ BLOCK          (2) WRITE BLOCK
 (3) SEND DISC COMMAND   (4) EXIT
STATUS: 00, OK,00,00

TRACK  : 12

SEKTOR : 01_

INPUTS IN HEX-CODE AND WITH TWO NUMBERS
```

[1] »Disc-Scanner 40«: keine Scheu vor den oberen Diskettenregionen!

Hauptmenü: Schaltzentrale des Programms

Zu den Funktionen des Auswahlmenüs:

<1> **Read Block:** Zunächst muß man sich entscheiden, ob man einen neuen Diskettenblock lesen (dann <F1> drücken!) oder sich den Inhalt des aktuellen Sektors im Puffer zeigen lassen will (<F3>).

- <F1> neuer Block: Das Programm fragt nach der gewünschten Spur- und Sektornummer. Achtung: Es sind nur hexadezimale Zahlen erlaubt, die man zweistellig eingeben muß (also notfalls mit führender »0«!). Eingabebeispiele:

Spur 1, Sektor 0:

```
TRACK: 01
SEKTOR: 00
```

Spur 13, Sektor 19

```
TRACK: 0D
SEKTOR: 13
```

Spur 18, Sektor 10:

```
TRACK: 12
SEKTOR: 0A
```

Spur 39, Sektor 15:

```
TRACK: 27
SEKTOR: 0F
```

Zunächst erscheinen nur die ersten 8 Byte eines Sektors (ebenfalls als Hexzahlen). Der Bildschirmteil rechts bringt die entsprechenden Bytes im Klartext (ASCII-Code). CHR$-Codes, die das Programm nicht identifizieren kann, erscheinen als Punkt. Kurzer Tipp auf <SHIFT> fördert weitere Byte-Reihen zutage und scrollt den Bildschirm nach oben, bis die Reihe ab $F8 erreicht ist (Abb. 2). Jetzt sind Sie automatisch im Editiermodus (der Editor-Cursor steht auf dem ersten Byte-Wert links oben ab Byte $88 und läßt sich wie gewohnt mit den Cursor-Tasten bewegen). Wollen Sie Daten schon in davorliegenden Pufferbereichen ändern, müssen Sie bei der Hexdump-Ausgabe <RUN/STOP> drücken. Zur Änderung des Sektorinhalts sind ebenfalls nur hexadezimale Werte erlaubt – die alten Bytes werden überschrieben, der neue Inhalt der gesamten Byte-Zeile mit <RETURN> übernommen.

```
DISC-SCANNER 40 (C) 1992 MARC HAWLITZECK
 (1) READ BLOCK          (2) WRITE BLOCK
 (3) SEND DISC COMMAND   (4) EXIT
STATUS: 00, OK,00,00
> 88: 52 41 4D 45 4E 55 45 20  RAMENUE
> 90: 4D 41 4B 45 52 00 00 00  MAKER...
> 98: 00 00 00 00 00 00 25 00  ......%.
> A0: 00 00 82 11 01 55 4C 54  .....ULT
> A8: 52 41 42 4F 4F 54 20 4D  RABOOT M
> B0: 41 4B 45 52 2B 00 00 00  AKER+...
> B8: 00 00 00 00 00 00 1C 00  ........
> C0: 00 00 82 10 05 55 4C 54  .....ULT
> C8: 52 41 4D 45 4E 55 20 54  RAMENU T
> D0: 4F 4F 4C A0 A0 00 00 00  OOL.....
> D8: 00 00 00 00 00 00 04 00  ........
> E0: 00 00 82 11 00 55 4C 54  .....ULT
> E8: 52 41 4C 4F 41 44 20 50  RALOAD P
> F0: 4C 55 53 A0 A0 00 00 00  LUS.....
> F8: 00 00 00 00 00 00 09 00  ........
LAST TRACK AND SEKTOR : 12 01
```

[2] Von Byte $00 bis $FF: Der gesamte Datenblock läßt sich editieren.

Kurzinfo: Disc-Scanner 40

Programmart: Diskettenmonitor
Laden: LOAD "DISC-SCANNER 40",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: berücksichtigt auch die oberen Spuren 36 bis 40
Benötigte Blocks: 20
Programmautor: Marc Hawlitzeck

Der Bildschirm mit der Byte-Liste läßt sich zwar nicht aufwärts scrollen, aber per <F7> kommt man wieder zum Blockbeginn (ab Byte-Nr. $00), mit <F1> kehren Sie ins Auswahlmenü zurück.
- <F3> Datenpuffer ansehen: bringt den zuletzt geladenen Sektorinhalt nochmals auf den Bildschirm und kann wie bei Funktion <F1> bearbeitet bzw. editiert werden.

Sektoreninhalte zurückschreiben

<2> **Write Block:** ...ermöglicht es, geänderte Blocks zurückzuschreiben, Sektoreninhalte in andere Blocks auf derselben Diskette oder auf andere Scheiben zu übertragen. Dazu müssen Sie den gewünschten Track und Sektor angeben. Zur besseren Orientierung meldet das Programm die Kennzahlen des zuletzt gelesenen Blocks. Anschließend springt »Disc-Scanner 40« automatisch wieder ins Auswahlmenü.

<3> **Send Disc Command:** Mit <F1> schickt man die bekannten DOS-Befehle (z.B. N:, S:, R: usw.) ans Laufwerk (ohne OPEN- und CLOSE) oder ruft per <F3> die integrierte schnelle Formatierroutine auf (Abb. 3):
- Taste <1>: Speed Format Track 1 bis 23 (=1 bis 35), benötigt 37 Sekunden,
- Taste <2>: Speed Format Track 1 bis 29 (=1 bis 41), dauert 43 Sekunden,

```
DISC-SCANNER 40 (C) 1992 MARC HAWLITZECK
 (1) READ BLOCK          (2) WRITE BLOCK
 (3) SEND DISC COMMAND (4) EXIT
STATUS:

1. SPEED FORMAT TRACK 1 - 23
2. SPEED FORMAT TRACK 1 - 29
3. SPEED FORMAT TRACK X - Y
```

[3] Drei Formatier-Varianten lassen keine Wünsche offen

- Taste <3>: Speed Format Track x bis y (frei wählbare Start- und Endspur, vor allem für die oberen Tracks nützlich!).

Vor dem Formatieren muß man entscheiden, ob das Laufwerk einen »Bump« ausführen soll (korrekte Justage des Schreib-Lesekopfs durch Anschlagen am Chassis). Das ist aber nur dann nötig, wenn Sie zuvor auf eine höhere Spur als 35 zugegriffen haben. Dann nämlich muß der Kopf auf alle Fälle neu positioniert werden. Geben Sie nun Disknamen und ID ein - die Diskette wird formatiert. Per Tastendruck kommt man wieder ins Auswahlmenü.

Selbstverständlich läßt sich auch das übliche DOS-Kommando zum Formatieren per <F1> aufrufen, z.B.:

```
COMMAND: N:TESTDISK,01
```

ID-Kennung des Hauptdirectory

Dann werden aber grundsätzlich nur die Spuren 1 bis 35 berücksichtigt, außerdem dauert's 105 Sekunden!

Falls Sie mit Funktion <3> nur bestimmte Tracks formatieren möchten (z.B. Nr. 36 bis 38) müssen Sie bei der Frage nach der ID-Kennung unbedingt den 2-Byte-Wert übernehmen, der für die 35 Stammspuren gilt - sonst erhalten Sie bei künftigen Zugriffen auf neuformatierte Spuren eine Fehlermeldung (Disk ID Mismatch). Die Routine formatiert immer inkl. angegebener Endspur!

<4> **Exit:** Damit verläßt man »Disc-Scanner 40« ohne Reset. Der Bildschirm erhält die Grundfarben zurück (wie nach dem Einschalten des C64). Jetzt läßt sich das Disk-Tool mit RUN erneut starten. Wer den Reset-Knopf gedrückt hat, verbannt den Basic-Start des Programms aus dem Speicher, kann aber mit SYS 35000 wieder einsteigen.

Programmhinweise

»Disc-Scanner 40« ist ein Assembler-Programm, das nach dem Laden zwar im Bereich $0801 (2049) bis $1B45 (6981) liegt, aber per C-64-Systemroutine zum Verschieben ganzer Datenbereiche ($A3BF) nach $88B8 (35000) bis $9BD1 (39889) transferiert wird.

Wer mit Hintergrund-, Rahmen- und Vordergrundfarbe des Bildschirms nicht einverstanden ist, kann die Werte vor dem Start mit RUN ändern. Achtung: Bei der Vordergrundfarbe gelten nicht die Farbcodes von 0 bis 15, sondern die entsprechenden CHR$-Werte lt. C-64-Handbuch, Anhang A (z.B Schwarz = CHR$(144):

Datenpuffer wird reserviert

```
POKE 2093,Hintergrundfarbe (0 bis 15)
POKE 2098,Rahmenfarbe (0 bis 15)
POKE 2129,Vordergrundfarbe als CHR$
POKE 2171,Vordergrundfarbe als CHR$
```

Wenn in der Programmbeschreibung vom Datenpuffer die Rede ist, handelt es sich um den 256 Byte großen Bereich von $8E70 (36464) bis $8F6F (36719) innerhalb von Disc-Scanner 40, den das Programm als Speicherplatz für den gelesenen Diskettensektor zur Verfügung stellt. Von dort wird er zum Editorbildschirm übertragen. Alle Änderungen auf dem Bildschirm registriert der Pufferbereich ab $8E70 ebenfalls, dessen Inhalt mit Funktion <2> (Write Block) wieder zur Diskette gesandt wird.

Ab und zu kann's vorkommen, daß nach einem Leseversuch die Meldung »Read Error« erscheint: In der Normalkonfiguration hätte der Schreib-Lesekopf Ihrer Floppy jetzt zu rattern und schlagen begonnen (Bump), um sich neu zu positionieren. Verschobene Köpfe in oberen Spuren richten sich nämlich nicht automatisch wieder ein! Disc-Scanner 40 unterdrückt den Kopf-Bump (lobenswert, weil materialschonend!). Abhilfe bringt hier Menüpunkt <3> (Send Command): die Floppy mit »I« neu initialisieren und nochmals den gewünschten Sektor aufrufen.

Diskmonitor für alle Spuren

Bei unseren Tests klappte die Arbeit mit Disc-Scanner 40 auf den Floppies 1541, 1541-II und 1571 (ebenso integriertes Laufwerk des C-128D-Blech im C-64-Modus!) problemlos.

Schwierigkeiten machte allerdings die 1581. Obwohl diese Floppy geradezu prädestiniert scheint, Daten in oberen Spuren abzulegen und zu verwalten. Das DOS 10.0 der 3½-Zoll-Diskettenstationen spielt aber nicht mit: Die Floppy läuft zwar an, aber der Bildschirm bringt z.B. in Spur 40, Sektor 3, (Directory-Block) nur Null-Bytes.

Wer nicht vom Pfad der Tugend abweichen will und mit den mysteriösen Spuren ab Nr. 36 nichts am Hut hat, kann Disc-Scanner 40 wie einen normalen Diskmonitor für Track 1 bis 35 verwenden. Auch ohne integrierte Directory-Funktion bietet er alle wichtigen Funktionen, um Disketteninhalte komfortabel zu manipulieren. (bl)

Disk Searcher – findet jedes Byte auf Diskette

Spürhund

Will man einen ganz bestimmten Text oder Byte-Wert in den 664 Sektoren einer Diskseite aufspüren, hilft auch ein noch so komfortabler Diskettenmonitor nicht viel. Doch »Disk Searcher« verbeißt sich in die schwarze Scheibe, bis er die gesuchten Daten am Wickel hat.

Ein Fallbeispiel: Sie besitzen eine Datendiskette zu einem Textverarbeitungsprogramm mit über 100 Einzeldateien. Irgendwo darin verbirgt sich eine Textpassage (Namen, Telefonnummern, Jahreszahlen usw.), die Sie ändern möchten. Mit einem Diskmonitor lassen sich einzelne Blöcke auf Diskette schnell und problemlos editieren oder ändern – wenn man weiß, in welchem Sektor die Daten stehen! Die andere Variante, die Textverarbeitung zu laden und alle Textdateien zu durchforsten, bringt nur dann etwas, wenn die Dateibezeichnungen nicht zu allgemein sind (z.B. TEXT 1, BRIEF 5 usw.). Aber was ist, wenn Sie zehn Datendisketten haben? Von der Zeit, die bei dieser Suchmethode draufgeht, wollen wir gar nicht reden.

Kurzinfo: Disk-Searcher

Programmart: Disketten-Scanner
Laden: LOAD "DISK-SEARCHER",8
Starten: nach dem Laden mit RUN
Besonderheiten: benötigt 45 Sekunden für den Check einer Diskettenseite und gibt Spur- und Sektornummern aus, in der Suchtext gefunden wurde. Kommt ohne Assembler-Routinen aus!
Benötigte Blocks: 5
Programautor: Manfred Müller

```
DISK - SEARCHER VON MANFRED MUELLER
STARTTRACK? 1
ENDETRACK? 35
JOKER? 255

CODE? 64'ER REDAKTION
CODE? DAS MAGAZIN
CODE? FUER COMPUTERFANS
CODE? E

DISK EINLEGEN
```

Alle Eingaben erledigt? Dann kann die Suche beginnen...

»Disk Searcher« durchforstet eine gesamte Diskettenseite in knapp 45 Sekunden. Bei den gesuchten Zeichenfolgen läßt sich der vom Floppy-DOS bekannte Joker <?> einsetzen.

Laden Sie das Utility mit:

```
LOAD "DISK-SEARCHER",8
```

und starten Sie mit RUN.

Das Programm fragt nach dem Bereich, in dem es herumschnüffeln soll (Start- und Endspur). Die Werte »1« und »35« sind voreingestellt, die man per <RETURN> übernehmen sollte. Mit vorangestelltem <$>-Zeichen sind auch hexadezimale Eingaben möglich (also $23 statt 35). Anschließend möchte das Utility wissen, welches Jokerzeichen verwendet wird. $FF (255) ist vorgegeben; es entspricht dem vom DOS verwendeten Fragezeichen <?>. $3F (63) bringt denselben Effekt. Wir empfehlen, die Voreinstellung zu übernehmen und mit dem Zeichen <?> zu arbeiten.

Komfortable String-Suche

Bei der nächsten Frage (Code:) kann man Bytes oder Zeichenketten eintragen, nach der das Programm suchen soll (mit <RETURN> bestätigen!). <E> beendet die Eingabe und fordert Sie auf, die gewünschte Diskette einzulegen, auf der Sie die Byte-Folge vermuten (Abb.).

Erneuter Tipp auf <RETURN> macht der Floppystation Beine: Der Schnüffler ist von der Leine. Blitzschnell meldet der Bildschirm Spuren und Sektoren (immer dezimal!), in denen das Programm den Suchstring entdeckt hat. Die Zahlen sollten Sie sich merken (besser: notieren). Laden Sie nun die Sektoren per entsprechendem Diskmonitor, ändern Sie diese nach Belieben und speichern die Blöcke zurück.

Wie sieht der Such-String aus?

Das Programm versteht unterschiedliche Sucheingaben, die aber die maximale Länge von 32 Zeichen nicht überschreiten dürfen:

- Dezimalzahlen: einen ASCII-Wert zwischen 0 und 255, z.B. »14« für CHR$(14), aber »49« für die Zahl »1« = CHR$(49),
- Hexadezimal: Solchen Zahlen muß man ein $-Zeichen voranstellen. Zur Verfügung steht der Bereich von $00 bis $FF. Beispiel: $0E für CHR$(14) oder $31 für die Zahl »1«,
- ASCII-Code: Damit forscht man nach Texten oder Zeichenketten im ASCII-Code (CHR$). Leiten Sie die Sucheingabe mit dem Apostroph <'> ein, sonst interpretiert »Disk Searcher« den ersten Buchstaben als Zahl. Achtung: Den Suchtext nie mit einem weiteren Hochkomma abschließen! Beispiel:

Suchbegriffe mischen

```
CODE? 'DISK SEARCHER VON MANFRED
CODE? E
```

sucht nach Tipp auf <RETURN> die Überschrift des aktuellen Menübildschirms von »Disk-Searcher« und findet den Text nach 45 Sekunden in Spur 11, Sektor 0 – vorausgesetzt, die Diskette zu diesem Sonderheft liegt im Laufwerk! Es wird grundsätzlich die gesamte Diskettenseite gecheckt.

Selbstverständlich läßt sich auch gemischter Suchcode eingeben, z.B.

```
CODE? 'DISK SEARCHER VON MANFRED
CODE? ' MUELLER
CODE? 13
CODE? $0D
CODE? 'STARTTRACK
CODE? E
```

bringt als Ergebnis ebenfalls Spur 11, Sektor 0.

Entsteht während der Suche ein Diskettenfehler (Read Error), bricht das Programm ab und gibt die entsprechende Meldung aus. »Disk Searcher« arbeitet mit der 1541 ebenso zusammen wie mit der 1571 (Einzellaufwerk oder intern im C128D). Die Nummern der entdeckten Diskettenblöcke werden in einem Floppy-Pufferspeicher abgelegt: Damit kann das Programm also nur maximal 128 gefundene Spuren und Sektoren ausgeben. (bl)

Fast Format – von 0 auf 664 in 11 Sekunden!

... im High-Speed-Modus

Das Floppy-DOS der 1541 braucht zum Formatieren einseitiger 5¼-Zoll-Disketten fast anderthalb Minuten. »Fast Format« schafft in der Zeit acht!

Bevor man Disketten zur Datenablage nutzen kann, müssen die schwarzen Scheiben in vorgeschriebene Spuren (Tracks) und Sektoren (Blocks) eingeteilt werden: Per entsprechendem Befehl wird das vom Floppybetriebssystem automatisch erledigt – pro Diskettenseite fast 80 Sekunden!

»Fast Format« macht der Floppystation Beine: Nach 11 Sekunden hat man eine funktionstüchtige Disk.

Laden Sie das Utility mit:

```
LOAD "FAST FORMAT",8
```

Das Hauptprogramm in Assembler (FF.OBJ) wird nachgeladen. Dann fordert Sie das Programm auf, die zu formatierende Diskette ins Laufwerk zu legen und <RETURN> zu drücken. Anschließend gibt man Diskettenname (Header), ID-Kennung und die Geräteadresse des zu aktivierenden Laufwerks ein (Standardfloppy 8 oder Zweitfloppy 9).

Wenn das Laufwerk zu rattern beginnt, ist's fast schon vorbei: Nach 11 Sekunden meldet sich der Bildschirm mit der Frage, ob man eine weitere Diskette formatieren will.

Ohne Basic-Programm »Fast Format« geht's auch:

```
LOAD "FF.OBJ",8,1
NEW
SYS 49152,N$,I$,D
```

N$ ist der Diskettenname, I$ die ID und D die Geräteadresse. Die Strings kann man auch im Klartext eingeben, z.B.:

```
SYS 49152,"64'ER SONDERHEFT","S1",8
```

Damit erzielt man denselben Effekt wie mit dem Basic-Programm. »Fast Format« funktioniert problemlos mit den Floppies 1541, 1541-II, 1571 und dem internen Laufwerk des C128D – allerdings nicht mit der 1581! (bl)

FASTFORMAT 11.0
by Jan Kusch
Diskette einlegen,
dann <RETURN>!

Diskname:64'ersonderh.83
ID-Kennung:s1

Die schnelle Formatier-Routine startet unmittelbar nach der Parametereingabe

Kurzinfo: Fast Format

Programmart: schnelle Formatier-Routine für 5¼-Zoll-Disketten
Laden: LOAD "FAST FORMAT",8
Starten: nach dem Laden mit RUN
Besonderheiten: formatiert in ca. 10 Sekunden
Benötigte Blocks: 10
Programmautor: Jan Kusch

Sir-Scan – enttarnt Diskettenfehler

Wanzenjagd

Fehlerhafte Disksektoren bringen Ärger: Wertvolle Dateien oder Programme lassen sich nicht mehr laden. »Sir-Scan« deckt »Bugs« (Wanzen) schonungslos auf!

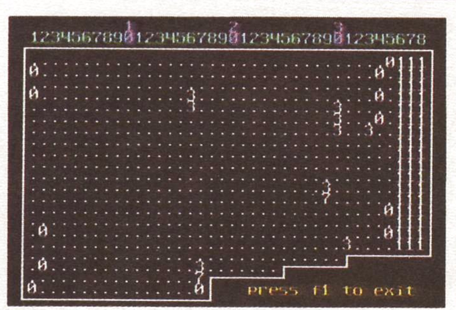
Unser Utility lokalisiert diverse Diskettenfehler, die laut 1541-Floppyhandbuch mit Nummern zwischen »20« und »29« gekennzeichnet werden.

- 20: Block-Header nicht gefunden,
- 21: Sync-Markierung nicht identifiziert,
- 22: Datenblock nicht korrekt gespeichert,
- 23: Prüfsummen-Fehler im Datenblock,
- 27: falsche Prüfsumme im Datenblock-Header.

Anschließend kann man per Diskmonitor versuchen zu retten, was zu retten ist: die Datenblöcke reparieren oder entfernen.

Kurzinfo: Sir-Scan

Programmart: Disketten-Utility
Laden: LOAD "SIR-SCAN",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: integrierte Funktion zur Überprüfung der Diskettendichte
Benötigte Blocks: 25
Programmautor: Christian Dombacher



Fatale Leserfehler: am besten Diskette neu formatieren!

Laden Sie das Utility mit:

```
LOAD "SIR-SCAN",8
```

Nach dem Start mit RUN wird der Assembler-Teil »I-SIR-SCAN« nachgeladen, dann erscheint die Scannerfläche. Die Zahlenleiste oben bringt die Spurnummern (1 bis 38!). Die Sektoren sind aus Platzgründen nicht numeriert: Die unterste linke Ecke des Scannerbildschirms bezeichnet Datenblock 0, Nr. 20 würde links oben stehen.

Die Menüfunktionen (vorher gewünschte Disk einlegen, dann entsprechende Taste drücken!):

<E> – **Error Scan:** Diskette auf Fehler prüfen. Sektoren, die »clean« sind, erzeugen einen Punkt auf dem Screen. Andernfalls erscheint in der entsprechenden Sektor-Koordinate eine der erwähnten Fehlernummern ohne vorangestellte »2« (Abb.). Beispiel: »3« ist »Read Error 23«.

<D> – **Density Scan:** ...gibt Auskunft über die Schreibdichte einer formatierten Diskette. Je höher die Zahl (von 1 bis 4), desto höher die Dichte!

Nicht selten entstehen Diskettenfehler durch die unterschiedliche Dichte einzelner Spuren oder Sektoren schon ab Werk. Wertvolle Datendisketten sollten Sie also nach dem Formatieren, aber vor der Datenspeicherung mit der Funktion <D> überprüfen! (bl)

C 64-Tools – Diskettenmagier für 1541- und 1581-Floppies

Der Manipulator

Übersichtliche Directories, Sektoreninhalte komfortabel ändern, kinderleichte Diskverwaltung mit der 1581 und ein integriertes Kopierprogramm: Das sind die Highlights dieses Softwarepakets nach dem Vorbild »PC-Tools« für IBM-kompatible PCs!

C64-Tools« unterstützt die Laufwerke 1541 und 1581 (1571 nur mit Einschränkungen!). Es bedient maximal vier Diskettenstationen der Geräteadressen 8 bis 11 und bringt nützliche Funktionen von Einzel-Utilities für die genannten Floppies unter einen Hut.

Die Hauptaufgabe des Programms läßt sich schnell auf einen Nenner bringen: Directories modifizieren und in neuer Form auf Diskette zurückschreiben. Nach der Arbeit mit C64-Tools sollte stets ein VALIDATE (Diskettenbefehl V) durchgeführt werden – vor allem, wenn Einträge im Inhaltsverzeichnis gelöscht wurden!

Laden Sie das Programm mit:

```
LOAD "C64-TOOLS",8
```

und starten Sie mit RUN.

Die Datei »Infotext« (File-Typ USR) wird nachgeladen. Dann können Sie einstellen, mit welchem Laufwerk Sie arbeiten möchten (Nr. 8 bis 11). Legen Sie die zu bearbeitende Disk in die Floppy. Falls Sie die C-128-Laufwerke 1570/1571 verwenden, weist Sie das Programm darauf hin. Das Laufwerk wird nun als 1541 eingestellt.

Das Directory erscheint in der Arbeitsfläche des Menübildschirms (Abb. 1). Die Statuszeile im oberen Screen-Bereich zeigt den Header der aktuellen Disk, unten finden Sie das Funktionsmenü. Die Optionen ruft man mit der farblich hervorgehobenen Buchstabentaste auf (Beispiele: <L>oesch, <S>ortier, s<I>chern, <F2>: freigeben usw.).

Bevor wir auf die Menüpunkte im einzelnen eingehen, zunächst Hinweise zu allgemeinen Programmfunktionen:

Manchmal verlangt das Tool Texteingaben (z.B. beim Umbenennen eines Eintrags oder bei der Eingabe eines Diskbefehls). Meist bringt der Bildschirm eine Vorgabe, beispielsweise den alten Namen des Eintrags. Drückt man eine Steuertaste (<CRSR>, <HOME>, <DEL>, <INST>), bleibt die Vorgabe bestehen und läßt sich ändern. Jede andere Taste löscht das Eingabefeld und schafft Platz für neuen Text: Alle Zeichen sind erlaubt, sogar <RETURN> (erscheint als inverses M!). Nur: Wer reverse Steuerzeichen erzeugen will, muß gleichzeitig die CTRL-Taste drücken. Also <CTRL RETURN>, oder: Bildschirm löschen (reverses Herz). Dann sind drei Tasten gleichzeitig nötig: <CTRL SHIFT HOME>.

Einige Zeichen sind nicht einmal per Tastatur zu erreichen. Dann öffnet man mit <CTRL ←> ein Eingabefenster, in das man beliebigen ASCII-Code (Hexzahl oder dezimal) eintragen darf. Dieses Window ist ebenfalls nützlich, wenn in Assembler-Dateien bestimmte Befehlsfolgen gesucht werden sollen (z.B. $A9 $05 für LDA #05). Zeichen lassen sich direkt aus einer ASCII- oder Zeichensatz-Tabelle wählen. Zahlenwerte, nach denen Sie gefragt werden, geben Sie genauso wie Texte ein.



[1] C64-Tools: Arbeitsbildschirm mit Funktionsmenü

[2] Vereinfachte Manipulationen von 3½-Zoll-Disketten: Subdirectories bei der 1581

Arbeitsbildschirm mit komfortablen Funktionen

Doch nun zu den Merkmalen des Menübildschirms:

Ein roter Balken markiert den ersten Eintrag des Directory. Der Balken läßt sich mit den Cursor-Tasten bewegen.

Die Statuszeile unmittelbar unter dem Directory zeigt vier Zahlen: Die beiden ersten geben Auskunft über die Anzahl der Blöcke und Einträge. Beispiel:

```
395/ 77 B/D
```

bedeutet: 395 Blöcke und 77 Einträge (Dateien) stehen im aktuellen Directory.

Um das Inhaltsverzeichnis zu modifizieren, muß man zunächst Einträge wählen. Da gibt's mehrere Möglichkeiten:

einzeln: Nach <RETURN> wird der aktuelle Eintrag gewählt, bzw. freigegeben,

Block: ...aktiviert man per <SHIFT RETURN>. Der aktuelle Eintrag ist nun der Anfang eines zusammenhängenden Blocks von File-Namen. Mit den Cursor-Tasten kann man das Blockende frei bestimmen.

Eigenschaften: Mit <+> aktivieren Sie die Dateiauswahl, per <-> gibt man Einträge frei.

Position: Bei jeder bisher genannten Auswahlmethode

C64-Tools: Texteditor

Man darf beliebige Texte eingeben, auch solche, die Steuerzeichen enthalten. Allerdings stehen nur die über Tastatur zu erreichenden Zeichen zur Verfügung. Das Eingabefenster <CTRL –> läßt sich nicht aktivieren! Die Textmenge ist auf den Umfang des Editorbildschirms (Abb. 4) begrenzt (19 Zeilen 40 Zeichen = 760 Byte).

Steuerfunktionen

<SHIFT INST/DEL>: ...aktiviert den Einfügemodus, bzw. schaltet ihn wieder aus. Bei Eingaben wird der folgende Text weitergeschoben bzw. überschrieben. Der aktuelle Modus (I oder O) erscheint am rechten Rand der ersten Statuszeile.

<F7>: ...setzt den Cursor aufs nächste Wort,

<F5>: ... aufs vorhergehende.

<RETURN>: Cursor zum Anfang der nächsten Zeile (Line Feed).

<SHIFT RETURN>: ...setzt das Absatzendezeichen (inverse Spitzklammer). Diese Endmarkierung gilt auch beim Einfügen von Text. Zweimal <SHIFT RETURN> beendet den Einfügemodus an aktueller Cursor-Position und ignoriert nachfolgenden Text.

</> für Trennvorschläge: ..., die Sie bereits im Text vorsehen sollten. Folgt mitten im Wort nach mindestens einem Kleinbuchstaben ein großer, wertet C64-Tools dieses Zeichen als neuen Silbenanfang und übernimmt ihn als Kleinbuchstaben. Beispiele:
BuchStaBen: Silbentrennung Buch/sta/ben,
C64Tools: läßt sich nicht trennen,
C64-Tools: C64-/Tools

<–> Bindestriche: ...werden ähnlich wie Trennvorschläge interpretiert. Will man das verhindern, muß man den Trennungsstrich (= Minuszeichen) zusammen mit <CTRL> drücken. Jetzt erscheint ein inverses Minus, das eine Trennung an dieser Position im Wort verhindert.

Beachten Sie dazu die Datei »Infotext« auf der beiliegenden Diskette zu diesem Sonderheft, die im Texteditor mit <F4> geladen wird!

C64-Tools: System- und Fehlermeldungen

Hier ein Auszug wichtiger Systemmeldungen mit Erläuterungen zur Fehlerursache:

Warnung! manipuliertes Directory! Ignorieren? (j/n): Die Directory-Blöcke werden von der 1541 und 1581 gewöhnlich in einer bestimmten Reihenfolge angelegt. Man kann mit C64-Tools die geänderte Reihenfolge ignorieren und die Default-Werte wieder herstellen (<J>) oder die vorgegebene Reihenfolge beibehalten (<N>). Dann lassen sich allerdings keine neuen Blöcke ans Directory hängen!

Achtung! Geloeschte Eintraege sind danach verloren! Trotzdem schreiben? (j/n): ...erscheint als Rückfrage vorm Sichern des geänderten Directory,

Diskettenfehler! Floppy xx antwortet nicht! Laufwerk mit der Nummer xx nicht angeschlossen oder nicht eingeschaltet.

Fehler im Directory: Das Directory ist entweder über mehrere Spuren verteilt oder erzeugt eine Endlosschleife: Der Folgezeiger des letzten Blocks wird auf einen bereits gelesenen Block gerichtet.

Directory voll! Sie haben versucht, mit den Funktionen TRENNER, TEXT oder UNTERDIRECTORY ERSTELLEN zusätzliche Einträge zu erzeugen, für die kein Platz mehr vorhanden ist.

Warnung! Inkompatible Typwandlung! Kann zu Problemen fuehren! Trotzdem ausfuehren? (j/n) ...erscheint beim Versuch, einen Dateityp sinnlos zu ändern. Kompatibel sind:
- DEL (nur mit sich selbst),
- SEQ, PRG, USR (untereinander)
- REL (mit sich selbst)
- CBM (mit sich selbst, nur 1581)
- Illegale (nur untereinander)

Kopieren ist leider nur mit 2 Laufwerken moeglich! Aufgrund mangelnden Speicherplatzes wäre ein File-Copy mit nur einem Laufwerk zu zeitraubend und arbeitsaufwendig (häufige Diskettenwechsel).

Es werden nur sequentielle Dateien kopiert! Sie haben versucht, auch andere Dateitypen als SEQ, PRG und USR zu kopieren (z.B. REL).

DEL, CBM und Dateien mit illegalem Typ koennen nicht bearbeitet werden! Solche Dateien lassen sich mit dem Hex-Editor weder bearbeiten noch durchsuchen.

Daten wurden geaendert! Bitte SICHERN oder VERWERFEN waehlen! Man wollte den Hex-Editor mit STOP verlassen, obwohl der Block geändert wurde! Drücken Sie entweder <F2> oder <F8>! Mit jeder anderen Taste geht's in den Editor zurück.

Unerlaubte Folgespur! ...entspricht der DOS-Meldung »Illegal track or sector«. Tritt bei der Funktion »Finden« auf.

Kein Platz fuer Unterdirectories! Sie wollten auf einer 3½-Zoll-Disk der 1581 ein Unterverzeichnis installieren, obwohl weniger als drei zusammenhängende Spuren jeweils 40 freie Blöcke enthalten (Vorschrift des 1581-DOS 10.0: mindestens 120 freien Sektoren zur Installation eines Subdirectories!)

werden die favorisierten Einträge ans Ende der Auswahlliste gehängt. Möchte man aber einen Eintrag an bestimmter Lage im Directory unterbringen, gibt man die gewünschte Positionsnummer ein. Es spielt keine Rolle, ob der Eintrag schon in der Auswahlliste erfaßt war oder nicht.

<L>oesch: (Löschen)

...mit Sicherheitsabfrage:

- <J>(Ja): ...löscht nur den gezeigten Eintrag und fragt den nächsten ab.
- <N>(Nein): Der aktuelle Eintrag wird verschont, das Programm fragt nach dem nächsten,
- <R>(Rest): ...tilgt den gezeigten und alle anderen gewählten Einträge.
- <K> (Keine): Der aktuelle und alle übrigen selektierten Dateien werden nicht gelöscht. Achtung: Nicht vergessen, die Disk nach dem Löschen zu validieren!

<S>ort.: (Sortieren).

Sind keine Einträge markiert, wird alles sortiert. Sonst bringt das Tool die gewählten Namen auf die Reihe und stellt sie an den Directory-Anfang.

Ein Untermenü bietet diese Sortierkriterien:

– Auswahlnummer: ...reiht markierte Einträge nach Positionsnummer. Damit erreicht man quasi manuell eine beliebige Reihenfolge im Directory.

– Größe: ...sortiert aufsteigend nach Blockumfang,

– Name: ...alphabetisch (gemäß ASCII-Tabelle),

– Typ: ...sortiert nach Dateityp (in dieser Reihenfolge: DEL, SEQ, PRG, USR, REL, CBM (nur 1581!), Illegal)

Bei mehreren Einträgen gleichen Typs sortiert sie das Programm untereinander nach Attributen: *TYP, *TYP<, TYP, TYP<. Zusätzlich muß man entscheiden, ob steigend oder fallend sortiert werden soll.

<E>intrag:

...erlaubt es, Dateinamen und deren Parameter zu manipulieren (Abb. 3):

– Größe: Damit stellt man beliebige Dateigrößen ein (von 0 bis 65 535 Blocks). Aber Vorsicht: C64-Tools geht davon aus, daß die Information über die Dateigröße korrekt ist! Die Zahl der belegten Blöcke im Hauptmenü ist fürs Programm bindend – auch, wenn's ein Fantasiewert ist! Die Möglichkeit, Blockzahlen zu ändern, soll vielmehr dazu dienen, falsche Angaben zu korrigieren. Wie die korrekte Größe einer Datei ermittelt wird, findet man in der Beschreibung zum Hex-Editor.

– Name: Beliebiger Text ist erlaubt – sogar Steuerzeichen dürfen zur Namensänderung verwendet werden!

– Typ: Wählen Sie den gewünschten Dateityp per Anfangsbuchstaben. Aber auch hier unsere Warnung: Unsinnige Angaben (z.B. aus einem Basic-Programm (PRG) eine SEQ-Datei zu machen oder eine überdimensionale Blockanzahl einzutragen) bringen sicher Heiterkeitserfolge, aber keine Vorteile. Nur durch Änderung des Dateityps wird z.B. aus einer PRG-Datei noch lange kein REL-File!

Zusätzlich kann man folgende Funktion aufrufen:

– Attribute: ...setzt spezielle Dateimarkierungen:
- <*> für nicht geschlossene Dateien,
- <(> hinter dem Dateityp: Schreibschutzkennzeichen für dieses File.

– Füllen: Alle gewählten Dateinamen lassen sich mit <SPACES> auf 16 Zeichen verlängern oder auf weniger kürzen (Leerzeichen löschen).

<T>renner

...setzt per <T> oder <SHIFT T> Trennstriche an aktueller Position des Cursor-Balkens.

Te<X>t

...funktioniert ähnlich wie der Trenner. Damit kann man beliebige Texte ins Directory einbauen. Das Editorfeld erscheint und fordert Sie per blinkendem Cursor auf, Text einzutragen. Mit <F1> wird er linksbündig, per <F3> zentriert hinter dem Cursor-Balken ins Directory eingefügt. Ist die Textpassa-

ge länger als 16 Zeichen (= eine Eintragszeile), wird sie automatisch in mehrere Textzeilen untereinander aufgeteilt. Beachten Sie unseren Textkasten, der die Editorfunktionen beschreibt.

Mit <F4> kann man ein Text-File laden, <F2> sichert das aktuelle. Die Datei heißt stets »Infotext«. Achtung: Das File »Infotext« auf der Sonderheftdiskette dient nur als Platzhalter und läßt sich nicht laden (falscher Dateityp: SEQ). Dazu muß man erst beliebigen Text im Editor eingeben und mit <F2> als USR-Datei speichern. Sichern und Laden bezieht sich immer aufs aktuelle Laufwerk und Directory.

s<I>chern

...schreibt das manipulierte Diskinhaltsverzeichnis nach einer Sicherheitsabfrage auf Diskette zurück.

Achtung: C64-Tools prüft nicht, ob die richtige Diskette im Laufwerk liegt!

<F>inde

...forscht nach maximal 32 Byte langen Zeichenfolgen in sequentiellen (SEQ, PRG, USR, z.B. Texte) oder relativen Dateien (REL). Alle gewählten Files werden durchsucht. Wurde die Zeichenkette entdeckt, ruft man den Hex-Editor zum Ändern auf oder setzt die Suche fort. Das Programm entdeckt auch Texte, deren Zeichenfolge das Ende eines Sektors überragen und teilweise in den nächsten hineinreichen.

<H>exedit

...aktiviert den integrierten Diskettenmonitor (Hex-Editor). Allerdings sind seine Funktionen gegenüber echten Diskmonitoren eingeschränkt: Es lassen sich weder Dateien mit illegalem File-Typ (z.B. $CF<) noch DEL-Dateien oder Unterdirectories (1581-Partitions CBM) bearbeiten!

Sektoren Byte für Byte

Wurden mehrere Einträge gewählt, muß man sich zunächst per Cursor-Tasten für einen entscheiden. Wer den Editor erstmals aufruft, braucht ein bißchen Geduld: bei der 1541 etwa 30 Sekunden, ca. 1 Minute bei der 1581. Die gesamte Diskette wird nämlich auf Folgeblöcke untersucht. Dann erscheint der Editorbildschirm mit den Hexzahlen des ersten Datenblocks. Das Menü im unteren Bildschirm bietet Optionen, die sich mit folgenden Tasten realisieren lassen:

<CLR/HOME> bzw. <SHIFT CLR/HOME:>: ...bringen den ersten bzw. letzten Block der gewählten Datei.

<B>locknummer: Geben Sie eine Zahl an: der Inhalt dieses Blocks erscheint als Hexdump auf dem Bildschirm. Achtung: Das ist eine interne physikalische Ordnungszahl und hat nichts mit der logischen Sektornummer auf der Disk zu tun! Mit dieser Funktion kann man z.B. die wirkliche Größe einer Datei erkunden. Gibt man z.B. 9999 als Blocknummer an, wird immer der letzte Datenblock auf den Screen geholt.

<CRSR rechts/links>: Da nur jeweils die Hälfte des Sektorinhalts auf den Editorbildschirm paßt, blättert man damit vorwärts bzw. zurück.

<CRSR aufwärts/abwärts>: ...scrollt zeilenweise den Hexdump des Blocks.

Edit: ...ruft die Editorfunktionen auf: Ein roter Eingabe-Cursor erscheint in der linken oberen Ecke des Hex-Datenfelds.

Jetzt kann man den aktuellen Sektorinhalt beliebig ändern und manipulieren - von den Blockfolgezeigern (die ersten beiden Bytes am Sektorbeginn) sollten Sie aber die Finger lassen! Bytes, die man nicht verändern darf, sind grün hervorgehoben.

Auch das Hex-Editormenü bietet komfortable Arbeitsfunktionen:

- **<F1>**: Wechsel zwischen Hex- und ASCII-Eingabe (= Text). Bei der ASCII-Eingabe läßt sich mit <CTRL←> das besagte Eingabefenster aufrufen, um Tastencodes als Dezimalzahl einzutragen.
- **<F2>**: ...zurück zur Blockauswahl. Alle bis dato gemachten Änderungen sind hinfällig; der Blockinhalt wird neu geladen.
- **<F8>**: ...speichert den geänderten Sektor auf Diskette. Man kommt zur Hex-Editor-Blockauswahl zurück.

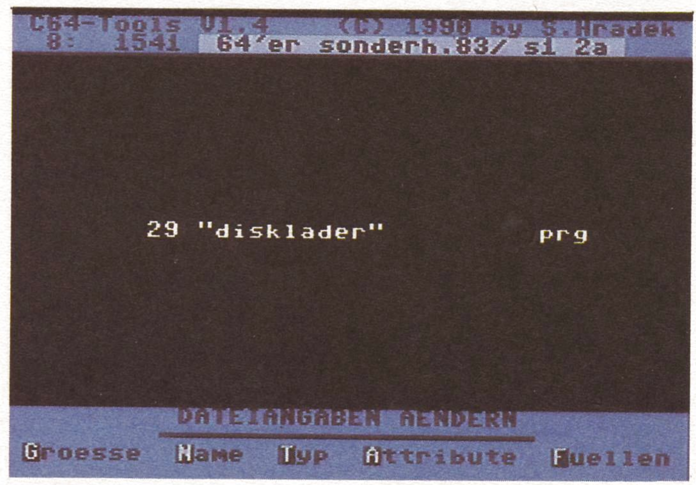
Mit <RUN/STOP> geht's wieder ins Hauptmenü.

<K>opier

...schaltet die integrierte File-Copy-Funktion ein. Alle Dateien sequentiellen Formats (SEQ, PRG, USR) werden problemlos übertragen. Die Kopierroutine berücksichtigt zwei Laufwerke. Die Geräteadresse des Ziellaufwerks muß sich von dem der Quellfloppy unterscheiden!

Befehl an Ziellaufwerk: Vor dem Kopieren kann man Diskettenbefehle an die Zielfloppy senden (<$> bringt z.B. das Directory) oder den Fehlerkanal abfragen (<CRSR aufwärts/abwärts>).

Anmerkung: Mit dieser Funktion läßt sich bei der 1581 zwar das aktuelle Verzeichnis wechseln (/0:, s. Beschreibung), ist aber nur dann sinnvoll, wenn die 3½-Zoll-Floppy als Ziellaufwerk fungiert. Bei der 1581 als aktuelle Diskettenstation (Nr. 8) könnte es zu Problemen kommen.



[3] Menüfunktion Eintrag: Dateiparameter beliebig ändern

Kurzinfo: C64-Tools

Programmart: Disk-Tool mit Kopierfunktion und Directory-Sorter
Laden: LOAD "C64-TOOLS",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: integriertes File-Copy-Programm nur für zwei Laufwerke. Die 1581 muß als Zielfloppy definiert sein.
Benötigte Blocks: 92
Programmautor: Stephan Hradek

Kopieren starten: Bevor Sie mit <K> das File-Copy starten, sollte man zuvor im Hauptmenü die gewünschten Dateien per rotem Cursor-Balken und <RETURN> markieren. Die gewählten Dateien werden blau unterlegt. Verzichten Sie darauf, Trennstriche mit 0 Blocks zu übertragen! Die Files werden in der Reihenfolge Ihrer Auswahl zum Ziellaufwerk geschickt.

Versucht man, eine Datei zu kopieren, deren Name auf der Zieldiskette bereits existiert (Fehlermeldung: »63, file exists«), gibt's vier komfortable Reaktionsmöglichkeiten:

- **<D>**:...löscht die alte Datei auf Disk und kopiert die gleichnamige neue,
- **<N>**:...verzichtet auf die Datenübertagung und macht mit der nächsten Datei weiter,
- **<A>**:...kopiert alle selektierten Dateien ohne Wenn und Aber. Eventuell gleichnamige Files auf der Zieldiskette werden automatisch durch gleichnamige der Quelldisk ersetzt.
- **<K>**:...überträgt alle gewählten Dateien mit Ausnahme solcher, deren File-Namen auf beiden Disketten identisch sind. Schutz vor unabsichtlichem Überschreiben!

Mit <RUN/STOP> brechen Sie die Kopieraktion ab. Obwohl die File-Copy-Funktion nicht die schnellste ist, klappt Sie auch mit einer 1581 als Ziellaufwerk einwandfrei!

<B>ef.

...sendet Floppy-DOS-Anweisungen ans aktuell definierte Laufwerk (S:, R:, N: usw. <$> zeigt das Directory). OPEN- und CLOSE-Befehle entfallen.

<D>isk

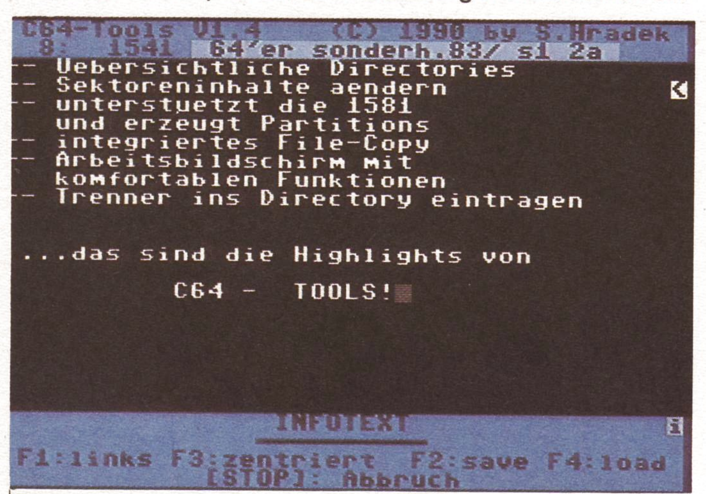
Damit lassen sich die Geräteadresse der angeschlossenen Laufwerke ändern, Diskettenwechsel anmelden oder das Directory neu lesen (falls Sie einen Editierfehler gemacht haben!).

<+> und <–>: ausw.

...bieten Möglichkeiten zur Dateiauswahl: <+> markiert Einträge mit bestimmten Merkmalen, <-> macht die Auswahl rückgängig. In den unteren vier Statuszeilen taucht ein weiteres Untermenü auf:

Größe: ...fragt nach der niedrigsten und maximalen Blockzahl pro File. Selektiert bzw. entfernt werden alle Einträge innerhalb des definierten Sektorenumfangs.

Name: Wir empfehlen, den vollständigen Dateinamen einzutragen. Außer <*> wird nämlich kein anderes Jokerzeichen des Floppy-DOS akzeptiert!



[4] Der integrierte Texteditor umfaßt eine Bildschirmseite

Typ: Per Tipp auf die Taste des entsprechenden Anfangsbuchstabens wählt man den Dateityp aus: Del, Seq, Prg, Usr, Rel, Cbm, Illegal.

<F1> trenner:

...hängt unmittelbar mit der Funktion <T> (Trenner) zusammen: Vor dem Einfügen ins Directory läßt sich das Aussehen des Abgrenzungsstrichs nach eigenen Wünschen verändern. Defaulteinstellung ist eine Zeile (16 Zeichen), die aus Minuszeichen besteht. Der vorgegebene Dateityp erscheint bei der 1581 als Typ DEL< (zusätzlich schreibgeschützt), bei der 1541 als illegaler Dateityp $CF< (im Directory erkennt man diese Manipulation an der Markierung ?<).

Mit den Tasten <N> und <T> lassen sich im Untermenü Name und Typ ändern.

<F2> freig.

...macht Ihre vorher getroffene Auswahl rückgängig: Alle markierten (blau unterlegten) Einträge werden freigegeben.

<F7> 1581

Das Spezialmenü für die 3½-Zoll-Floppy 1581 erscheint – allerdings nur dann, wenn sie vorher per Funktion <D> (Disk) als aktuelles Laufwerk eingestellt wurde!

Jetzt kann man sich beliebig durch die Directories einer 1581-3½-Zoll-Diskette hangeln. Die wichtigste Erweiterung des Standard-DOS 10.0 der 1581: Ab sofort kann man sich im Pfad auch **aufwärts** bewegen – es ist also nicht nötig, ständig erst das Stammverzeichnis (Root-Directory) aufzurufen, um von oben nach unten ins Unterverzeichnis zu kommen!

</> **Dir runter:** (darunterliegendes Directory aufrufen). Dazu muß der aktuelle Eintrag den Dateityp CBM besitzen und der in C64-Tools registrierte Name muß mit dem der Diskette identisch sein. Man kann also nicht zu einem Directory wechseln, dessen Name mit C64-Tools geändert und dies nicht auf Diskette vermerkt wurde. War alles in Ordnung, kommt das gewählte Directory auf den Bildschirm.

<.> **Dir rauf:** (nächsthöheres Inhaltsverzeichnis). Das übergeordnete Directory erscheint (der Punkt soll an den Befehl »cd ..« der PC-Betriebssysteme MS-DOS und Unix erinnern).

<↑> **Wurzel:** ...aktiviert das Hauptverzeichnis (Root-Directory, von engl. Root = Wurzel),

Zu den Unterverzeichnissen (Subdirectories) gibt's zwei Funktionen:

Init: ...ist nur dann nötig, wenn man die Floppy zwischenzeitlich ausgeschaltet oder die Diskette gewechselt hat.

Erstellen: Damit lassen sich komfortabel beliebige Unterverzeichnisse zum aktuelllen Directory einrichten. Nach <E> bringt der Bildschirm eine Übersicht aller Spuren der 3½-Zoll-Diskette im Laufwerk (Abb. 2). Alle gekennzeichneten Spuren (<+>, <-> oder <$>) gehören zum aktuellen Directory. <+>-Tracks sind komplett frei und stehen für Unterdirectories zur Verfügung.

Ein Unterverzeichnis braucht jedoch Parametereinstellungen:

– Größe: Von 120 Blocks (minimalste Größe für Unterverzeichnisse) bis zu einem bestimmten Maximalumfang (wird von C64-Tools automatisch ermittelt) darf man jeden Wert dazwischen eingeben: Die Zahl wird automatisch auf die nächste, durch 40 teilbare Zahl aufgerundet. Diese Bedingung ist für 1581-Unterverzeichnisse unverzichtbar. Per <CRSR aufwärts> überträgt man die Maximalgröße »1200« ins Eingabefeld, <CRSR abwärts> ermittelt den Maximalumfang ab aktuell gewählter Spur. Wenn nichts zusammenpaßt, (zu wenig freie Sektoren auf Disk), sollten Sie mit <RUN/STOP> wieder ins 1581-Menü zurückkehren und es nochmals (mit kleineren Werten) versuchen,

– Spur: Damit ist die Startspur gemeint. <CRSR aufwärts/abwärts> bestimmt den vorherigen bzw. nächsten passenden Track. Die Funktion ist übrigens ständig aktiv im 1581-Menü (auch ohne Optionswahl).

– Eintrag: ...zeigt den vorgegebenen Namen des Unterdirectory, der sich per Tastatur beliebig ändern läßt – lediglich die Zeichen <*> und <?> sind verboten!

– Dir: Damit definiert man den Namen des Subdirectory vor dem Formatieren (Vorgabe: »### ohne Namen ###«).

– ID: Mit einer zweistelligen ID-Kennung nach Wahl ist der Header fürs Unterverzeichnis komplett!

Nach diesen Vorbereitungen darf man nun mit <F1> das Verzeichnis anlegen. Anschließend geht's wieder zurück ins 1581-Spezialmenü.

C64-Tools ist nach dem Vorbild der MS-DOS-Software »PC-Tools« für Diskettenlaufwerke und Festplatten IBM-kompatibler PCs entstanden. Die Programmfunktionen, gepaart mit 100prozentiger Benutzerfreundlichkeit (Bedienungsfehler werden mit entsprechenden Hinweisen abgefangen, s. Textkasten) stehen denen des PC-Vorbilds in nichts nach!

1581-Besitzer, aufgepaßt: Auf der Sonderheftdiskette gibt's das separate Basic-Programm »Randomfiles«. Nach dem Laden und Start mit RUN muß man eine formatierte 3½-Zoll-Disk ins 1581-Laufwerk, Geräteadresse 9, schieben. Auf Tastendruck erzeugt das Demoprogramm 296 Testfiles mit unterschiedlichen Dateitypen. die sich nach Belieben manipulieren lassen (sortieren, Trenner einfügen usw.). Drei Beispieldateien haben wir ebenfalls auf die beiliegende Disk gespeichert.

(bl)

Ultraformat V2 – Disketten mit zwei Directories

Aktion Eichhörnchen

Sparen lautet die Devise: Mit »Ultraformat V2« holen Sie aus jeder 5¼-Zoll-Diskette 95 zusätzliche Datensektoren heraus!

Der Bildschirm meldet nach dem Formatieren einer 5¼-Zoll-Scheibe: »664 Blocks free«. Das sind 35 Spuren mit einer bestimmten Anzahl Sektoren zwischen 17 und 21 (s. Floppyhandbuch).

Dabei liegt in den höheren Spuren 36 bis 41 ein Block-»Eldorado«, das bei normalem Formatieren nie berücksichtigt wird: Unser Programm hilft Floppy-DOS auf die Sprünge!

Speichern oder kopieren Sie das Programm auf eine separate Arbeitsdiskette und laden Sie es mit:

```
LOAD "ULTRAFORMAT V2",8
```

dann starten Sie mit RUN.

Achtung: Das Tool funktioniert nur mit den Floppies 1541 und 1541-II (nicht 1570, 1571 oder 1581!).

Nach dem Initialisieren des Laufwerks erscheint das Arbeitsmenü (Abb. 1). Es bietet drei Optionen, die man mit den Cursor-Tasten anwählt und per <RETURN> aufruft (der aktuelle Menüpunkt wird revers angezeigt):

Ganze Diskette formatieren: ...wirkt wie der normale Formatierbefehl des DOS, aber: Zusätzlich werden die Spuren 36 bis 41 bearbeitet! Dazu muß man den Disknamen und die ID-Kennung eingeben und <RETURN> drücken. Wenn die Floppy fertig ist, sind zunächst nur die Spuren 1 bis 35 formatiert. Jetzt wartet das Programm auf Ihre Entscheidung, ob die höheren Spuren formatiert und zwei Umschaltprogramme auf Diskette geschrieben werden sollen: »Diskswitch C000« und »Diskswitch 033C«. Achtung: Verwenden Sie dazu nicht die beiliegende Sonderheftdiskette (dort sind die beiden Dateien nämlich schon gespeichert!). Zwei Speichermodi stehen zur Auswahl (Aufruf per entsprechender Taste):

- <1> als Programmdateien (sie belegen zwei Blöcke auf der Tool-Disk),

```
*****  Ultraformat  V2.0  *****

Ganze Diskette formatieren
Nur Spur 36-41 formatieren
Umschaltprogramme ersetzen

Auswahl mit Cursortasten und  Return

  disk-name? track36-40

  disk-id  ? bl
```

[1] Das Arbeitsmenü: Pro Diskette lassen sich 24 320 Byte zusätzlicher Speicher generieren!

- <2> in der Directory-Spur 18. Dann bleibt die übrige Speicherkapazität auf Disk unberührt, allerdings können dann im Disketteninhalts-Verzeichnis maximal nur 128 Einträge vermerkt werden (statt 144).

Nur Spur 36 bis 41 formatieren: ...erfüllt den eigentlichen Zweck des Programms: Bereits beschriebene Disketten lassen sich so bis aufs letzte Byte nutzen, ohne Daten in den Spuren 1 bis 35 zu zerstören! Eine Voraussetzung muß die Disk aber erfüllen: Es sollten noch mindestens zwei Blöcke auf der Scheibe (oder im Directory) frei sein! Denn: Auch bei diesem Formatmodus kann man nicht auf die »Diskswitch«-Dateien verzichten!

Umschaltprogramme ersetzen: ...kommt nur in Frage, wenn Sie die alte Version von »Ultraformat« (64'er-Sonderheft 25) besitzen und die Dateien austauschen möchten. Sonst können Sie diesen Menüpunkt getrost vergessen!

Zweites Directory in den Hyper-Spuren

Um die neuen Spuren 36 bis 41 mit den gewohnten DOS-Anweisungen zu erreichen, gibt's die beiden identischen »Diskswitch«-Maschinenprogramme. Sie liegen nach dem Laden lediglich in verschiedenen Speicherbereichen:

```
LOAD "DISKSWITCH C000",8,1 (bzw. 033C)
```

Per SYS Startadresse werden sie eingeschaltet (C000 = SYS 49152, 033C = 828). Ab sofort erhält die <RESTORE>-Taste eine weitere Funktion: Sie schaltet zwischen beiden Bereichen um (Achtung: Beim ersten Tipp auf <RESTORE> werden automatisch die unteren Spuren aktiviert!). Also: Erst der nächste Druck auf die Umschalttaste zeigt, was sich auf den mit »Ultraformat V2« manipulierten Disketten geändert hat: Klammheimlich wurde nämlich beim Formatieren der oberen Spuren in Track 36 ein zusätzliches Directory generiert, das 95 freie Blöcke anzeigt (Abb. 2)! Die ID-Kennung lautet: »Oben«.

```
READY.
LF"$",8

SEARCHING FOR $
LOADING
READY.
L

0 "TESTDISK5         " S1 2A
1    "DISKSWITCH C000"   PRG
1    "DISKSWITCH 033C"   PRG
662 BLOCKS FREE.
READY.
LF"$",8

SEARCHING FOR $
LOADING
READY.
L

0 "TESTDISK5         " OBEN
0    "NICHT VALIDIEREN"  PRG
95 BLOCKS FREE.
READY.
```

[2] Mit dem zweiten Disketten-Inhaltsverzeichnis in den oberen Spuren kann man wie gewohnt arbeiten

Wenn Sie in diesem Umschaltmodus bleiben, funktionieren die üblichen Floppybefehle einwandfrei: Programme laden, speichern, Directory in den Speicher holen usw. Erneuter Tipp auf <RESTORE> bringt wieder den normalen Modus des Floppy-DOS zurück. Wichtiger Hinweis: Verzichten Sie fürs obere Directory auf den DOS-Befehl VALIDATE!

Die beiden Basic-Programme »Switch U-O.Bas« (Umschaltung von unten nach oben) und »Switch O-U.Bas« (von oben nach unten) simulieren die Funktionen der Assembler-Datei »Diskswitch« und lassen sich in eigenen Programmentwicklungen einsetzen. (bl)

Kurzinfo: Ultraformat V2

Programmart: Formatprogramm
Laden: LOAD "ULTRAFORMAT V2",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: Diskette erhält zweites Directory in den höheren Spuren und wird per komfortablem Umschaltprogramm aktiviert.
Benötigte Blocks: 25
Programmautor: Ralph Gauer

Double-Format – zwei Disketten gleichzeitig!

Im Doppelpack

Wenn nur das nervtötende Formatieren neuer Disketten nicht wäre! Mit »Double-Format« wird's zum Vergnügen – zum Mega-Fun, wenn man zwei Floppystationen besitzt!

Schneller geht's kaum: Unser Tool formatiert einseitige 5¼-Zoll-Disketten in weniger als 10 Sekunden! Am meisten Zeit sparen Besitzer zweier Laufwerke 1541 (1541-II): Beide Diskettenstationen lassen sich gleichzeitig bedienen.

Laden Sie unser Super-Tool mit:

```
LOAD "DOUBLE-FORMAT",8
```

und starten Sie es mit RUN.

Achtung: Vergewissern Sie sich, daß vor dem Start beide Floppystationen (Geräteadressen 8 und 9) angeschlossen und eingeschaltet sind, sonst wird die Zweitfloppy vom Utility nicht erkannt!

Das Hauptmenü (Abb.) bietet vier Optionen, die man mit den entsprechenden Funktionstasten aktiviert:

<F1> **Single Format**: ... bearbeitet nur die Diskette im Standardlaufwerk (Gerätenr. 8). Geben Sie den Diskettennamen und die ID-Kennung (zwei- bis fünfstellig!) an. Anschließend kann man bestimmen, ob das Floppy-DOS während des Formatierens ein VERIFY ausführen soll. Legen Sie die gewünschte Disk ins Laufwerk und drücken Sie <RETURN>. Nach dem Formatieren meldet sich der Computer mit dem Fehlerstatus (ideal ist »00, ok, 00, 00«!). Per Tastendruck geht's zurück ins Hauptmenü.

<F3> **Double-Format**: ... entspricht Menüpunkt 1, allerdings wird nun auch gleichzeitig die Diskette in Laufwerk 9 formatiert (Name und ID sind identisch).

<F5> **Directory**: ... bringt das Inhaltsverzeichnis der aktuellen Diskette auf den Bildschirm.

<F7> **Quit**: ... beendet das Programm mit einem Reset.

Probieren Sie's aus: Mit »Double-Format« bearbeitet man eine Zehner-Packung Disketten in knapp einer Minute! (bl)

double-formatter by the sir in 1988
f1 : single-format
f3 : double-format
f5 : directory
f7 : quit

Im Hauptmenü wählt man, ob man nur eine oder zwei Disketten formatieren will

Kurzinfo: Double-Format

Programmart: Formatier-Tool
Laden: LOAD "DOUBLE-FORMAT",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: arbeitet auch mit zwei Laufwerken gleichzeitig. Achtung: funktioniert nur mit 1541-Floppies!
Benötigte Blocks: 7
Programmautor: Christian Dombacher

Track-Copy II – Disketten-Backup in 90 Sekunden

Datenschleuder

Schneller geht's kaum noch: »Track-Copy II« dupliziert eine vollständige Diskettenseite in anderthalb Minuten!

Rechnen Sie's nach: In einer Stunde lassen sich mit unserem Utility etwa 30 einseitige Disketten kopieren.

Zusätzlich ist »Track-Copy II« voll kompatibel: Es arbeitet mit der älteren 1541 ebenso wie mit den neuen Versionen 1541-II. Auch die C-128-Laufwerke 1570 und 1571 machen im C-64-Modus mit Geräteadresse 8 keine Probleme.

Laden Sie das Copy-Tool (Abb.) mit:

```
LOAD "TRACK-COPY II",8
```

und starten Sie es mit RUN.

Legen Sie die Quelldiskette ins Laufwerk. Beantworten Sie folgende Fragen (<N> für »No«, sonst <Y> für »Yes«):

- small BAM: ... kürzt den Kopiervorgang ab. Diese Option wirkt sich aber nur bei weniger als 200 Blöcken aus.
- Lesezugriff wiederholen (Repeat): Bei »Y« versucht es das Laufwerk bis zu fünfmal.
- Verifying on: vergleicht Daten beim Kopieren.
- Format & Copy: ... ist mit »Y« zu antworten, wenn man eine blanke, unformatierte Zieldisk verwendet. Dann muß zusätzlich zum Disknamen auch die ID vermerkt werden. <RETURN> übernimmt automatisch die Einträge.

Jetzt fordert Sie das Programm im Kopierverlauf auf, bis zu dreimal Quell- und Zieldiskette zu wechseln (Source und Destination). Zur Meldung »Kopie fertig« (Copy is done):

- <*> (ist voreingestellt) aktiviert das nächste Kopiermenü,
- <$> ruft das Backup-Directory auf (zeigt Spur, Sektor, Blocks, Dateiname). Nach <RETURN> erscheint »Order:«: Mit <*> geht's zum Kopiermenü,
- jedes andere Tastaturzeichen: ... kopiert die nächste Diskette mit den gleichen Menüeinstellungen.

Beenden läßt sich das Programm per Reset oder <RUN/STOP RESTORE>. Danach sollten Sie die Floppy initialisieren oder vor weiterer Benutzung kurz ausschalten. (bl)

Track-Copy II by S.I.G
small BAM? n
repeat read sector?n
VERIFYING on? n
format & copy?j
Disk-Name: copydisk
Source-disk in drive 8

Im Kopiermenü sind die notwendigen Parameter einzutragen

Kurzinfo: Track-Copy II

Programmart: Disketten-Backup
Laden: LOAD "TRACK-COPY II",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: funktioniert mit allen C-64- und C-128-Floppytypen außer 1581!
Benötigte Blocks: 18
Programmautoren: C 64/C 128 Special Interest Group (S.I.G.)

Wizcopy V1.0 – drei Utilities in einem Programm

Allround-Talent

Kopieren ist nicht alles: »Wizcopy« sortiert auch Directory-Einträge oder gibt die Listen in Etikettenform auf Drucker aus!

Bei diesem Disk-Tool kann man gar nichts falsch machen: Wizcopy meldet jeden Bedienungsfehler! Eins muß man aber beachten: Während der Aktivierung der Programmfunktionen muß der seriell angeschlossene Drucker ausgeschaltet bleiben, da das Tool beim Laden und Speichern Fast-Load-Save-Routinen benutzt. Schalten Sie den Drucker erst unmittelbar vor dem Drucken ein (z.B. beim Directory-Druck).

Laden Sie das Kopier-Tool mit:

```
LOAD "WIZCOPY V1.0",8
```

und starten Sie mit RUN.

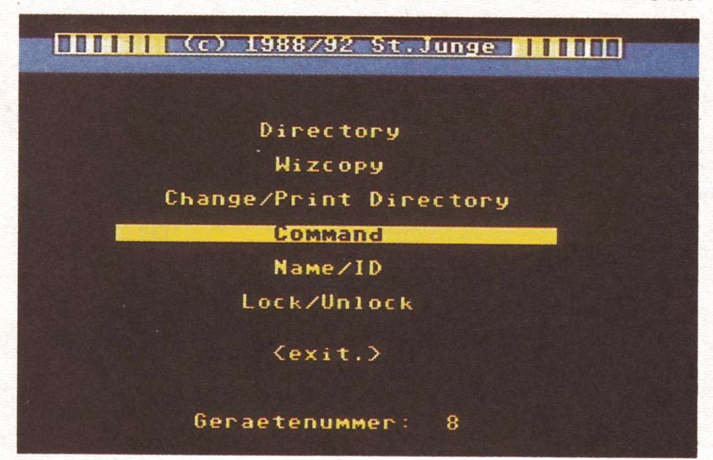
Das Hauptmenü (Abb. 1) bietet Optionen, die per Cursor-Tasten ausgewählt und mit einem reversen Balken gekennzeichnet werden. Falls nötig, können Sie zunächst mit der **Taste <G>** eine Floppygeräteadresse zwischen 8 und 11 einstellen.

Per <RETURN> lassen sich die Menüpunkte aufrufen:

Directory: ...zeigt das Inhaltsverzeichnis der aktuellen Diskette im Laufwerk. Die Bildschirmausgabe läßt sich jederzeit mit einer beliebigen Taste stoppen (festhalten!). <RUN/STOP> bricht die Anzeige ab und kehrt ins Hauptmenü zurück.

Wizcopy: ...ruft die Kopierroutine für einzelne Dateien auf. Bevor man die Option aktiviert, sollte die Quelldiskette bereits im Laufwerk liegen: Das Directory wird gelesen und angezeigt (Abb. 2). Der erste Dateiname ist mit einem blauen Cursor-Balken unterlegt. Die Cursor-Tasten zeigen jetzt nur im Untermenü Wirkung (Bildschirm oben):

File kopieren: Per <F5> (nach oben) und <F7> (unten) setzt man den blauen Auswahlbalken auf den gewünschten File-Namen und markiert die Datei mit <RETURN>. Die linke Spalte zeigt die Auswahlreihenfolge.



[1] Das Hauptmenü von Wizcopy: Funktionwahl per Cursorbalken

Start: Sind alle Dateien markiert, aktiviert man die Kopieraktion. Die gewählten Files werden gelesen, im Datenpuffer abgelegt und auf dem Bildschirm mit dem Buchstaben L gekennzeichnet. Achtung: »Wizcopy« akzeptiert PRG-, SEQ-, aber auch USR-File-Typen (keine REL-Files!). Die Blockanzahl einer Einzeldatei darf »210« nicht übersteigen!

Bei der Meldung »Diskette wechseln und Space drücken« muß man die Zieldiskette einlegen und auf die Leertaste tippen: Die Dateien werden nun übertragen (neue Kennzeichnung: LS). Anschließend kann man sich entscheiden, ob man dieselben Dateien nochmals auf eine andere Diskette kopieren (j) oder lieber ins Hauptmenü zurückkehren will (n).

Neu wählen: ...macht die bisher getroffene Dateiauswahl rückgängig: Man muß von vorne beginnen.

Menü: zurück zum Bildschirm mit der Hauptauswahl.

Change/Print Directory: Das ist die umfangreichste Funktion von »Wizcopy«. Erneut wird das Directory der aktuellen Diskette im Laufwerk auf den Bildschirm geholt. Im oberen Screen erscheint ein neues Untermenü (Abb. 3) und die Anzahl der gelesenen Dateien. Der untere Bereich zeigt die Dateinamen in der Reihenfolge, in der sie sich auf Diskette befinden. Der blaue Cursor-Balken kann wie gehabt mit <F5> bzw. <F7> nach oben oder unten bewegt werden.

INS Line: Damit läßt sich der im Fenster gezeigte Trennstrich (Dateityp DEL) per <RETURN> an der Position des blauen Balkens ins Directory einfügen.

Position: Ein Tipp auf <RETURN> transportiert jetzt den gewählten Dateinamen ins Line-Fenster und entfernt ihn aus dem Directory. Nach erneutem <RETURN> wird der Name an gewünschter Stelle wieder ins Directory eingefügt (als wär's ein Trennstrich!).

Delete: ...löscht den File-Eintrag aus dem Inhaltsverzeichnis. Achtung: Um die BAM zu korrigieren, sollte man anschließend die VALIDATE-Funktion aufrufen (s. Hauptmenüpunkt »Command«).



[2] Im Kopiermenü werden die gewünschten Dateien markiert

Kurzinfo: Wizcopy V1.0

Programmart: Disketten-Tool
Laden: LOAD WIZCOPY V1.0",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: integrierte Directory-Sortierfunktion. Druckt Inhaltsverzeichnisse in Kleinschrift, die sich auf die Diskettenhülle kleben lassen.
Benötigte Blocks: 42
Programmautor: Stephan Junge

DEF Line: Wer mit dem Aussehen des Trennstrichs nicht einverstanden ist, kann ihn hier im Line-Window nach Belieben editieren (z.B. Sternchen, durchgezogene Linie usw.). Alle inversen Steuerzeichen (Farbcodes oder Cursor-Bewegungen) werden ebenfalls akzeptiert.

Print: Frühestens jetzt sollten Sie Ihren Epson-kompatiblen Drucker (serieller Anschluß oder entsprechen-

des Hardware-Interface) einschalten. Erst jetzt läßt sich nämlich eine Liste in Etikettengröße auf dem Drucker ausgeben, die man z.B. ausschneiden und auf die Diskhülle kleben kann. Die Routine berücksichtigt alle Änderungen, die Sie bisher schon mit dem Directory angestellt haben: neue Trennlinien eingefügt, File-Namen gelöscht usw.

Geben Sie das Datum ein (<RETURN> tut's auch!) und wählen Sie zwischen ein-, zwei-, dreispaltigem Druck oder Etikettenausgabe. Selbstverständlich können Sie auch jederzeit wieder ins Hauptmenü zurück, wenn Sie es sich anders überlegt haben. Wer mit dem Ausdruck Schwierigkeiten hat: s. Hinweise zur Druckeranpassung!

Change: ...bietet viele Möglichkeiten, Directory-Einträge zu manipulieren. Wieder werden die Funktionen in einem Untermenü aufgerufen:
- Name: Im Editierfenster erscheint der durch den blauen Balken unterlegte Dateiname und darf nach <RETURN> beliebig geändert werden. Erneuter Tipp auf <RETURN> übernimmt die Änderungen in der Directory-Liste.
- Typ: wechselt die DOS-Kennzeichnung für die Dateiart. Zur Festlegung des neuen Typs genügt es, die Tasten mit den Anfangsbuchstaben zu drücken: <P> = PRG, <S> = SEQ, <U> = USR, <R> = REL und <D> = DEL. Selbstverständlich sollten Sie Typenbezeichnungen nicht willkürlich ändern (Dateistruktur beachten!), sonst ist die neue Bezeichnung sinnlos.
- Blocks: stellt die Anzahl der Datenblöcke pro Datei auf einen gewünschten Wert ein (0 bis 65 535), der beim Laden des Directory angezeigt wird. Beliebt ist z.B. die Methode, alle Blockzahlen eines Directory auf »0« zu setzen. Allerdings: In

```
 Position             INS Line          Print
                      Position          Change
                      Delete
                      DEF Line          Load
 58 files                               Save

 0   ----------------   del             Menu

Nr   Blocks        Name                 Type
3    0             tools                usr
4    0                                  usr
5    37     ultramenue maker            prg
6    28     ultraboot maker+            prg
7    4      ultramenu tool              prg
8    9      ultraload plus              prg
9    0      ----------------            usr
10   91     c64-tools                   prg
11   3      infotext                    usr
12   8      testdatei 168               prg
13   4      testdatei 165               seq
14   10     testdatei 128               usr
15   41     randomfiles                 prg
16   0      ----------------            usr
17   16     sir-scan                    prg
```

[3] Option Change/Print Directory: Inhaltsverzeichnisse sortieren und ausdrucken

der BAM (Blockbelegunsplan) bleibt alles beim alten. Sie zeigt bei der Meldung »Blocks free« die korrekte Zahl.
- Close: schließt noch geöffnete Dateien auf Diskette, die nach einem Fehler beim Speichern mit dem Stern <*> gekennzeichnet sind.
- Lock/Unlock: ...schützt Dateien vor unbeabsichtigtem Löschen. Als Kennzeichnung erscheint hinter dem Dateityp eine Spitzklammer »<«. Wählt man bereits markierte Dateien mit dieser Funktion, geschieht das Gegenteil: Die Schutzkennung wird aufgehoben.

Load: ...holt das Inhaltsverzeichnis der Diskette,

Save: ...schreibt das geänderte Directory auf Diskette zurück,

Menu: ...bringt Sie wieder ins Hauptmenü.

Command: ...ruft im unteren Bildschirmteil eine Eingabezeile auf, in die man die gewohnten DOS-Befehle einträgt: S (= Scratch), N (= New), I (=Initialize) usw. An der verbindlichen Schreibweise lt. Floppyhandbuch hat sich nichts geändert – lediglich die OPEN- und CLOSE-Anweisungen entfallen, die man in Basic 2.0 stets eingeben muß.

Name / ID: ...bietet die Möglichkeit, Disknamen und ID-Kennung ohne Neuformatieren zu ändern. Man darf auch reverse Steuerzeichen verwenden. Die ID läßt sich auf insgesamt fünf Zeichen ausdehnen. <RETURN> beendet die Eingaben und schreibt den neuen Header auf Diskette zurück. Drückt man <RETURN> bei leerem Eingabefeld, übernimmt das Programm die alten Werte.

Lock / Unlock: Im Gegensatz zur gleichnamigen Menüfunktion bei der Option »Change« wird hier nicht eine einzelne Datei, sondern eine gesamte Diskettenseite gegen Überschreiben oder Löschen geschützt (als hätte man die Diskettenkerben seitlich mit einem Schreibschutzaufkleber versehen!). Auch die DOS-Anweisung VALIDATE bleibt ohne Wirkung, lediglich hartes Formatieren mit Angabe einer ID-Kennung erlaubt Schreibzugriffe auf die Diskette. »Wizcopy« selbst umgeht diesen Softwareschutz: Auch per »Lock« schreibgeschützte Disketten lassen sich manipulieren.

Wählen Sie »Unlock«, um die Sperre wieder aufzuheben.

<exit.>: ...beendet das Programm ohne Reset. Mit RUN läßt es sich erneut starten.

Editorprogramm zur Druckeranpassung

Gefallen Ihnen die Bildschirmfarben von »Wizcopy« nicht? Macht der Drucker Probleme, z.B. mit dem Zeilenvorschub? Dann laden Sie das Dienstprogramm:

```
LOAD "WIZCOPY-EDITOR",8
```

Legen Sie die Diskette mit »Wizcopy V1.0« ins Laufwerk und starten Sie mit RUN.



[4] Diskettenname und iD lassen sich komfortabel in separaten Datenfeldern ändern

Nach Tastendruck muß man Fragen z.B. nach Drucker- und Sekundäradresse beantworten oder kann die Bildschirmfarben und die des Cursor-Balkens ändern. Da »Wizcopy V1.0« beim Directory-Druck mit verschiedenen Schriftarten arbeiten (z.B. komprimierte oder gedehnte Schrift), sollten Sie im Druckerhandbuch nachschlagen und gegebenenfalls die Werte an Ihren Drucker anpassen. Für die meisten Epson-kompatiblen Drucker sind aber die Parameter goldrichtig eingestellt (Beispiel: Star NL 10), die man aber trotzdem nochmals eintragen muß (<RETURN> alleine genügt nicht!). »Esc« schreibt man im Klartext. Es entspricht dem in vielen Druckerhandbüchern erwähnten Steuerzeichen CHR$(27), dann folgen die weiteren Werte als Dezimalzahl. Für »Drucker initialisieren« muß die Eingabe z.B. so aussehen:

```
IHRE EINGABE: ? ESC 64
```

Anschließend wird die alte Wizcopy-Version auf der Diskette durch die neue ersetzt (das dauert seine Zeit).

Wizcopy V1.0 ist eines der kompaktesten Disketten-Tools für den C64, das Ihnen eine Menge Tipparbeit bei der Bedienung der Floppystation erspart. (bl)

Supercopy – bis zu 32 Dateien in einem Rutsch!

Multi-Transfer

»Supercopy« trägt sein hochtrabendes Prädikat zu Recht: Nahezu der gesamte freie Speicherplatz des C 64 dient als Datenpuffer – da muß selbst manches Disketten-Backup-Programm passen!

Das Kopier-Tool ist ein Meister im Ausnutzen freien RAMs: Knapp 60 000 Byte des C-64-RAM werden als Kopierspeicher genutzt. Das klappt aber nur, wenn man alle RAMs unterm Betriebssystem-ROM ausreizt und das Utility selbst äußerst speichersparend programmiert: »Supercopy« erfüllt diese Voraussetzungen, indem es auf Grafik-Schnickschnack bei der Bildschirmausgabe völlig verzichtet. Es funktioniert mit der 1541, 1570 und 1571 – allerdings nur bei einem Laufwerk. Richten Sie sich also auf »Handbetrieb« ein (= Diskettenwechsel)!

Funktionswahl im Hauptmenü

```
**** SUPER COPY 1541 ****
(C) BY N.MANN & D.WEINECK

1. DIRECTORY
2. KOPIEREN
3. FORMATIEREN
4. SCRATCH
5. VALIDIEREN
6. ENDE

BITTE WAEHLEN SIE :
```

Supercopy: Kopier-Tool für ein Laufwerk, mit immensem Datenpuffer

Kurzinfo: Supercopy

Programmart: Kopierprogramm für Einzeldateien und Single-Laufwerk
Laden: LOAD "SUPERCOPY",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: transferiert maximal 234 Blöcke bzw. 32 Dateien in einem Durchgang. Arbeitet mit den Floppies 1541, 1541-II, 1570 und 1571.
Benötigte Blocks: 12
Programmautor: N. Mann & D. Weineck

Laden Sie das Copy-Tool mit:

```
LOAD "SUPERCOPY",8
```

und starten Sie mit RUN.

Das Hauptmenü (Abb.) bietet sechs Auswahlmöglichkeiten, die man per entsprechender Zahlentaste aufruft:

<1> **Directory**: ...bringt das Inhaltsverzeichnis der aktuellen Diskette im Laufwerk mit Geräteadresse 8. Ein Tastendruck ruft das Menü wieder auf.

<2> **Kopieren**: ...aktiviert den Urzweck des Programms. Sie werden aufgefordert, die Quelldiskette einzulegen. Per <SPACE> erscheinen die Einzeldateien im Directory. Ob es in den Kopierspeicher übernommen werden soll, entscheidet ein Druck auf die Tasten <J> oder <N>. <Pfeil links> fungiert als Escape-Taste und bricht die Auswahl ab, kehrt aber ins Hauptmenü zurück (das gilt für alle Menüpunkte!). Es gibt keine andere Funktion, die Auswahl zu unterbrechen: Sie müssen die gesamte Liste ablaufen lassen, auch wenn Sie nur ein File kopieren möchten.

Sind alle Dateien gewählt, erscheint ein Untermenü (Optionen ebenfalls per Zahlentaste aktivieren!):
- 1. Directory,
- 2. Validieren: Die aktuelle Diskette wird aufgeräumt (Validate),
- 3. Formatieren: ...falls Sie vorher vergessen haben, die Zieldiskette zur Datenaufnahme vorzubereiten.
- <SPACE>: ...setzt die Kopieraktion fort.

Die markierten Dateien werden gelesen (Reading...) und im Kopierspeicher abgelegt. Übersteigt die Auswahl die magischen Grenzen von 234 Blöcken oder 32 Files, macht das Programm zunächst hier Schluß.

Auf mehrere Disketten kopieren

Legen Sie jetzt die Zieldiskette ins Laufwerk. Wählen Sie, ob alle Files fortlaufend (<1>) auf ein- und dieselbe Diskette oder einzeln (<2>, auf verschiedene Disketten) kopiert werden sollen. Bei <2> erscheint erneut das genannte Untermenü. Außerdem läßt sich das zuletzt gelesene Programm per Taste <4> nochmals auf eine andere Diskette schreiben. Der Bildschirm bringt den Namen der nächsten Datei, damit Sie die gewünschte Zieldiskette einlegen können.

Das Ende des Kopierdurchgangs signalisiert »Kopie fertig«. Supercopy merkt sich aber, ob schon alle gewählten Dateien kopiert wurden und fordert Sie gegebenenfalls auf, erneut die Quelldisk ins Laufwerk zu schieben. Sind alle Dateien auf der Zieldiskette gespeichert, kehrt man ins Hauptmenü zurück.

<3> **Formatieren**: ...löscht eine Zieldiskette zur erneuten Datenaufnahme komplett. Geben Sie den Disknamen und die ID ein (mit Komma getrennt).

<4> **Scratch**: Die Auswahl der zu löschenden Dateien läuft genauso ab wie bei Option <2> (Kopieren). Vor dem Löschen kommt eine Sicherheitsabfrage: Per Leertaste rottet man nun die markierten Dateien aus.

<5> **Validieren**: ...sollten Sie vor allem nach der Löschfunktion <4> wählen: Die Diskette wird aufgeräumt und die BAM aktualisiert.

<6> **Ende**: ...beendet das Programm ohne Reset. Man kann es mit RUN neu starten.

Wie die meisten bekannten File-Copy-Programme für den C64 muß man auch bei Supercopy eine Einschränkung machen: REL- und USR-Dateien werden nicht akzeptiert – das Programm macht nach entsprechender Meldung (»Überspringen oder neuer Versuch?«) mit der nächsten Datei weiter. Relative Files z.B. belegen noch »Side-Sektoren« mit

Unerlaubte Dateitypen

wichtigen Daten (Länge, Blockverbindungen usw.), die mit üblichen File-Kopierprogrammen sehr kompliziert zu lesen und auf die Zieldiskette zu übertragen wären. Außer Backup-Utilities, die nicht einzelne Files, sondern Sektor für Sektor pro Diskseite übertragen, ist uns nur ein File-Copy bekannt, das keine Scheu vor REL- bzw. USR-Dateien zeigt: »Magic Copy« im 64'er-Sonderheft 62. Dennoch: Dieser Schönheitsfehler mindert die komfortablen Disk-Funktionen von Supercopy keineswegs – vor allem, weil diese Dateitypen (abgesehen von Geos-Disketten) äußerst selten vorkommen! (bl)



Formatieren der Supertracks 36 bis 41, gelöschte Dateien wieder zurückholen, ein komfortables Fastload-Menü: Mit unserer Sammlung kurzer Utilities macht die Arbeit mit den Commodore-Laufwerken Spaß!

DOS-Operationen im Hintergrund

Werden Floppybefehle in den Zeilen eines Basic-Programms (z.B. zum Laden oder Speichern von Dateien) implementiert, führt sie das Laufwerk anstandslos aus – aber die Tastatur und andere Computerfunktionen stehen während der Datenübertragung nicht zur Verfügung.

Unser Utility bindet sich in den Interrupt ein: Alle DOS-Befehle wie LOAD, SAVE und VERIFY laufen quasi im Hintergrund ab und behindern Sie nicht in Ihrer Programmarbeit (z.B. bei der Dateneingabe oder Entwicklung eines neuen Basic-Programms)!

Laden Sie das Maschinensprache-Programm mit:

```
LOAD "IRQ-DOS $CE00",8,1
```

Die Funktionen der Mini-Basic-Erweiterung aktiviert man mit SYS 52736.

Obwohl sich auf dem Bildschirm sichtbar nichts ändert, stehen ab sofort neue Basic-Anweisungen zur Verfügung, die man im Direkt- oder Programmodus einsetzen kann:

!L »Dateiname«, Start: ...lädt eine Datei oder ein Programm an den Bereich »Start«. Beispiel:

```
!L "HIRES-GRAFIK",8192
```

holt das Bild »Hires-Grafik« ab Adresse $2000 in den Computer.

S »Dateiname«, Anfang, Ende: ...sichert den Speicherbereich zwischen »Anfang« und »Ende« als PRG-Datei auf Diskette. Beispiel:

```
!S "VIDEO",1024,2023
```

speichert den aktuellen Textbildschirm.

!V »Dateiname«, Start: ...vergleicht den Speicherbereich ab »Start« mit dem Daten-File »Dateiname« auf Diskette.

!E: ...bedeutet »Ende« (bricht die laufende Operation ab).

Zum Abschluß zwei Basic-Demos zu den komfortablen Programmfunktionen:

```
10 REM SPEICHERN UND LADEN DES VIDEO-RAM
20 IF A=0 THEN A=1: LOAD "IRQ-DOS $CE00",8,1
30 SYS 52736
40 !S "SCREEN",1024,2023
50 POKE 198,0: WAIT 198,1: REM TASTENDRUCK?
60 !L "SCREEN",1024
```

oder: Laden einer Datei (z.B. Assembler) innerhalb eines Basic-Programms, während gleichzeitig Daten über die Tastatur eingegeben werden:

```
10 REM LADEN WAEHREND DES PROGRAMMABLAUFS
20 IF A=0 THEN A=1: LOAD "IRQ-DOS $CE00",8,1
30 SYS 52736
40 !L "DATEI",49152
50 PRINT "GEBEN SIE DATEN EIN:"
60 INPUT D$
 ...usw.
```

Achtung: Beim LOAD-Befehl wird immer an die angegebene Adresse »Start« geladen (auch wenn diese nicht mit der auf der Disk eingetragenen übereinstimmt! (Jörg Brokamp/bl)

Byte für Byte enthüllt: das Floppybetriebssystem

DOS und DOS – das ist noch lange nicht dasselbe: Wir haben es aufgegeben, mitzuzählen, wie oft Commodore schon das Betriebssystem ein und desselben Floppytyps geändert hat: Die DOS-Version (Disk Operating System) der z.B. älteren 1541-C unterscheidet sich speziell in Einsprungadressen wichtiger Routinen von der neuen Fassung der 1541-II: Interessanterweise tragen dennoch beide dieselbe Versions-Nr: V2.6.

Noch gravierender war's bei der 1571 (DOS V3.0): als Single-Floppy gab's im Laufe der Monate zwei unterschiedliche Versionen auf dem Markt, die – äußerlich gleich ausgestattet – wiederum nicht kompatibel mit dem DOS V3.0 der internen Floppy des C128D (Plastik) waren! Das DOS V3.1 des C128D (Blechgehäuse) unterscheidet sich in derart vielen Adressen, daß man getrost von einem neuen Floppybetriebssystem sprechen kann!

Wie nicht anders zu erwarten: Auch die DOS-Version der neuen 1581 (s. S. 4) unterscheidet sich vom Betriebssystem der älteren 3½-Zoll-Floppy, die 1986/87 kurzfristig auf dem Markt war – Gott sei Dank nur in Kleinigkeiten!

Prinzipiell ist gegen die Neukonzeptionierung der Laufwerks-Betriebssysteme nichts einzuwenden: Fehler und Bugs wurden berichtigt, manchmal auch der Komfort gesteigert – nur: Super-Tools wie Kopier- oder Disketten-Manipulationsprogramme, die tief ins Floppy-ROM einstiegen und ausgiebig von Betriebssystem-Routinen Gebrauch machten, laufen plötzlich nicht mehr mit den DOS-Versionen der neuen Floppystationen (weil z.B. Einsprungadressen an andere Stellen verlagert wurden).

Findige C-64-Anwender wissen sich zu helfen: Sie brennen sich EPROMs mit dem jeweils passenden Betriebssystem. Je nach Bedarf läßt sich der Mikrochip durchs EPROM ersetzen. Wer fit in Assembler ist und über die Floppy-ROM-Routinen Bescheid weiß, findet sicher noch ein freies Speicherplätzchen, um eigene Maschinensprache-Routinen einzubauen, diese im EPROM-Code zu verankern und so die Floppybedienung noch freundlicher zu gestalten!

Wie kommt man an die 32 KByte des Floppybetriebssystems ran? Ganz einfach: per Anweisung Memory-Read (M-R). Dieser Befehl liest jedes Byte, das sich im Floppyspeicher versteckt hält: ROM oder RAM.

Laden Sie unser Demoprogramm in Basic 2.0 mit:

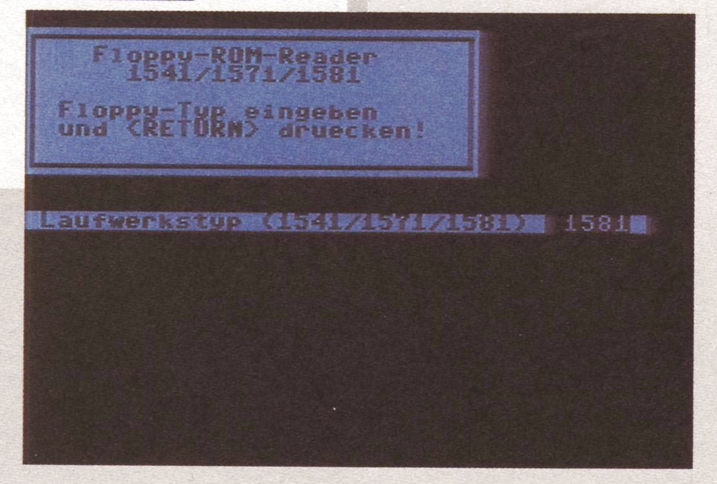
```
LOAD "FLOPROM-READER",8
```

Legen Sie nun eine formatierte Arbeitsdiskette mit mindestens 130 freien Blocks ins Laufwerk und starten Sie mit RUN.

Das Programm funktioniert mit allen Floppies des Typs 1541, 1571 und 1581, die auf Geräteadresse 8 eingestellt sind. Geben Sie den gewünschten Laufwerkstyp ein (also z.B. »1541«) und bestätigen Sie mit <RETURN>. Es müssen per FOR-NEXT-Schleife 32 768 Byte aus dem Floppy-ROM (von

$8000 bis $FFFF) geholt werden: Das geht in Basic nicht im Handumdrehen. Richten Sie sich also darauf ein, daß der Computer knapp 13 Minuten belegt ist!

Nach getaner Arbeit meldet sich der C64 mit der Frage nach dem Dateinamen für die Floppy-ROM-Datei (z.B. »ROM 1541-II«), die dann auf Diskette gespeichert wird und für die weitere Verarbeitung per Maschinensprachemonitor oder EPROM-Brenner zur Verfügung steht (Abb. 1). Die empfangenen Daten speichert der Computer ab Adresse $2000 (8192) bis Speicherstelle $9FFF (40159) im C-64-RAM. Sie müssen also $6000 (24576) addieren, um die echte Speicherstellennummer des Floppy-ROM zu erhalten.



[1] FlopROM-Reader: Nach Eingabe des Dateinamens wird das gelesene Floppy-ROM als Datei gespeichert.

Sollte es bei manchen Floppytypen dennoch Probleme geben (erkennbar an der zerstörten Bildschirmausgabe nach dem Lesen des Floppy-ROM), liegt's an den Floppy-Reset-Befehlen »UI+« (1541) und »UJ:« (1571/1581) in den Programmzeilen 10 bzw. 11. Wird diese Anweisung nach dem Einschalten des Laufwerks bei der aktuellen Computersitzung erstmals ausgeführt, erzeugt er intern die Meldung »73, DOS-System xx« und verteilt die Belegung der freien Datenpuffer anders - die Bytes werden per M-R-Anweisung nicht korrekt im Computer-RAM ab $2000 gespeichert!

Es gibt einen Trick, dies zu umgehen: Geben Sie unmittelbar nach dem Einschalten der Floppy oder zumindest vor dem Start von Floprom-Reader folgende Basic-Anweisung im Direktmodus ein:

```
OPEN 15,8,15,"UI+": CLOSE 15: REM 1541
```

bzw. für die Floppies 1571 und 1581:

```
OPEN 15,8,15,"UJ:": CLOSE 15
```

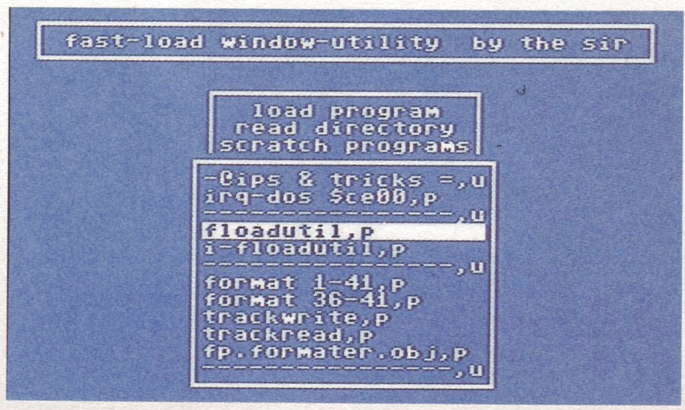
Außerdem sollten Sie folgende Programmzeilen ändern:

```
241 FOR I=128 TO 255: FOR J=0 TO 255
245 PRINT#15,"M-R",CHR$(J)CHR$(I)CHR$(1)
256 GET#15,X$
```

Noch ein Hinweis: Unsere Redaktion hat versucht, den ROM-Chip einer Single-Floppy 1571 nach dem Auslesen mit einem EPROM des DOS V3.1 vom C128D (Blech) zu bestücken, um die komfortableren Funktionen des DOS 3.1 in der internen Floppy des C128D (Blech) auch mit dem Einzellaufwerk zu nutzen. Aber: Vergessen Sie's, beide Floppystationen besitzen zu unterschiedliche Hardware-Konfigurationen! Bei identischen Floppytypen (zwei 1541-II, Single-1571 usw.) gibt's aber beim EPROM-Tausch der dazugehörenden Betriebssysteme keine Probleme. (bl)

Universelles Diskmenü mit Schnellader

Benutzeroberflächen zum raschen Laden von Programmen auf Diskette sind äußerst in: Per Knopfdruck Programm wählen, laden und starten. Der Nachteil: Manche Lade-Oberflächen sind überdimensional umfangreich (z.B. Geos) oder gelten nur für den jeweiligen Diskinhalt (z.B. der Disklader unserer Sonderheftdisketten).



[2] Floadutil: Programme per Tastendruck aus dem Directory laden oder löschen

Unser Ladeprogramm geht einen Schleichweg: Es holt die aktuellen Directory-Einträge blitzschnell in den Speicher und stellt eine zwar einfache, aber nicht minder komfortable Benutzeroberfläche zur Verfügung.

Laden Sie das Utility mit:

```
LOAD "FLOADUTIL",8
```

und starten Sie mit RUN.

Nach dem Entpacken und Laden des Assemblerteils »i-floadutil« meldet sich das Tool mit dem Menübildschirm (Abb. 2). Die Optionen werden mit <CRSR aufwärts/abwärts> gewählt und per <RETURN> aktiviert (vorher gewünschte Disk einlegen!):

load program: ...blendet ein neues Bildschirmfenster ein, dessen Inhalt per Cursor-Tasten nach oben oder unten scrollt: das Directory der aktuellen Diskette. Hinter dem Dateinamen erscheint der File-Typ (p, s, u, r). Bewegen Sie den inversen Cursor-Balken aufs gewünschte Programm und drücken Sie <RETURN>: Der Lade-Speeder tritt in Aktion, holt das Programm mit dem Faktor 20 (!) in den Speicher und startet es sofort. Achtung: Aus der Ladefunktion gibt's keinen Ausstieg (z.B. mit <RETURN> oder einer anderen Taste). Nach dem Laden des gewünschten Programms befindet sich »Floadutil« nicht mehr im Speicher und muß vor der Ladeaktion des nächsten Programms erneut in den Speicher geholt werden.

read directory: Die Bildschirmausgabe ähnelt der Funktion »Load« und bringt das Disketteninhaltsverzeichnis: Programme kann man allerdings keine aktivieren. Nach <RETURN> kommt man wieder ins Hauptmenü.

scratch programs: ...holt ebenfalls das Directory. Per Cursor-Balken und <RETURN> muß man alle Dateien markieren, die gelöscht werden sollen: Vor dem Dateinamen erscheint der Stern <*>. Erneuter Tipp auf <RETURN> macht die Auswahl wieder rückgängig. Scrollen Sie nun das Directory bis zur letzten Datei durch: Sie finden dort die Auf-

forderung »take selection«. Wenn Sie den Cursor-Balken darauf setzen und mit <RETURN> bestätigen, verabschieden sich die mit dem Stern markierten Dateien aus dem Directory. Anschließend kehrt man ins Hauptmenü zurück.

quit menu-loader: ...beendet das Utility mit einem Reset.

Aufgrund der verwendeten Schnelladeroutine ist die Kompatibilität des Tools eingeschränkt: Es funktioniert nur mit der 1541 und 1541-II. Keine Chance hat man mit der 1571 und 1581. (Christian Dombacher/bl)

Alles ist relativ!

Die meisten selbstprogrammierten Dateiverwaltungen benutzen sequentielle Dateien (SEQ, PRG, USR). Die Daten werden der Reihe nach hintereinander abgelegt, diese Einteilung höchstens durch spezielle Sortierroutinen (z.B. Bubble-Sort, Quick-Sort usw.) neu vergeben. Um nun an bestimmte Datensätze einer sequentiellen Datei ranzukommen, bleibt nur eins: Die gesamte Datenmenge muß in den Computer geholt und nach dem Suchbegriff durchforstet werden. Bei umfangreichen Dateien kann das eine kleine Ewigkeit dauern! Was würden Sie machen, um aus einer langen Buchreihe das zehnte herauszuziehen? Ganz sicher nicht die davorstehenden neun Bücher einzeln in die Hand nehmen und beiseitelegen, bis Sie beim zehnten Schmöker angelangt sind - sondern mit festem Griff eben dieses Buch aus dem Regal nehmen! Das entspricht dem Prinzip der relativen Dateiverwaltung (File-Typ REL): direkter Sofortzugriff auf jeden beliebigen Datensatz innerhalb einer umfangreichen Datei (z.B. Adressen). Und das wichtigste: Sie lagern die Datei vollständig auf Diskette aus: Eine REL-Datei kann dann also ohne weiteres aus mehr als 600 Blocks pro Diskseite bestehen - für ein SEQ-File sind solche Werte illusorisch, den es passen nicht mehr als maximal ca. 150 Blocks in den Basic-Arbeitsspeicher des C64, der auch noch Platz für die vom Programm benutzten numerischen und Stringvariablen bieten soll.

Durch integrierte Systemanweisungen unterstützt Sie Ihre Floppy mustergültig bei der Einrichtung von REL-Dateien. Sie ist aber auch auf Ihre Hilfe angewiesen: bevor man eine relative Datei einrichtet, muß man entscheiden, wieviel Platz jedem Datensatz (Record) zugestanden wird (die Summe aller Eingabezeichen, die man für die einzelnen Datenfelder verbraucht): Die Länge eines Records darf 254 Byte nicht überschreiten. Außerdem werden Floppyzugriffe und -anweisungen stets innerhalb des Befehlskanals (= Sekundäradresse 15) übermittelt. Der muß grundsätzlich als erste Programmanweisung geöffnet werden:

```
OPEN 1,8,15
```

Legen wir als Demobeispiel die REL-Datei »Reldat« mit der Record-Länge von 100 Byte auf einer leeren, formatierten Arbeitsdiskette an. Das Syntax-Schema lautet:

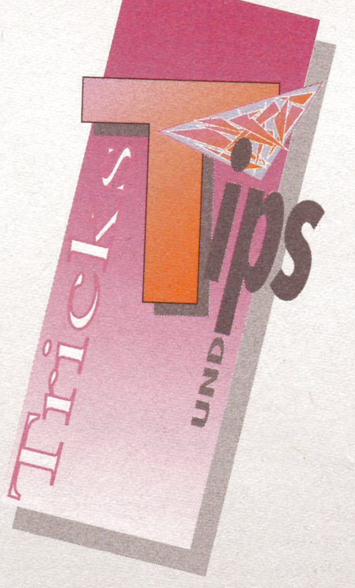
```
OPEN FN,GN,SA,"RELDAT,L,"+CHR$(RL)
```

Die Parameter bedeuten im Klartext:
- FN: logische Dateinummer (0 bis 255, außer 1 in unserem Beispiel),
- GN: Geräteadresse (z.B. Nr. 8),
- SA: Sekundäradresse - hier sind Werte zwischen 2 und 14 erlaubt!
- L: Dateityp (L = relative Datei)
- RL: Record-Länge (1 bis 254).

Für Laufwerk 8 könnte die Anweisung so aussehen:

```
OPEN 2,8,2,"RELDAT,L,"+CHR$(100)
```

Beachten Sie, daß die eingestellte Record-Länge für alle weiteren Programmoperationen bindend ist! Geben Sie beim späteren Öffnen einer bestehenden REL-Datei (z.B. zum Lesen) eine andere Zahl an, wird der Datenzeiger falsch positioniert. Ergebnis: nur Schrott, aber nie und nimmer der Inhalt des gesuchten Datensatzes!



Bei REL-Dateien greift das Laufwerk immer gezielt auf einen bestimmten Record zu. Welcher gemeint ist, muß der Floppy mit dem Positionierbefehl mitgeteilt werden - über den geöffneten Befehlskanal. Die Syntax-Vorschrift von P (Zeiger positionieren):

```
PRINT#1,"P"+CHR$(FN)
+CHR$(LO)+CHR$(HI)
+CHR$(BY)
```

Erläuterung zu den Parametern:
- FN: logische Dateinummer der REL-Datei,
- LO: Low-Byte der gewünschten Record-Nummer,
- HI: High-Byte,
- BY: Nummer des Zeichens im Datensatz, ab dem ein Schreib- oder Lesezugriff erfolgen soll.

In Basic 2.0 übernimmt der Interpreter für uns die Aufbereitung der Zahlen für den Mikroprozessor, die wir stets als Dezimalwert eingeben können. Greifen wir aber direkt ins Floppybetriebssystem ein, bleibt uns die Mühe nicht erspart, dem Floppyprozessor korrekte Werte zu liefern: Jede größere Zahl als 255 muß im Low-High-Byte-Format übergeben werden. Bauen Sie diese Umrechnung vor den Positionierungsbefehl ein (am besten als Unterprogramm):

```
870 REM *** FLOPPY-ZEIGER POSITIONIEREN ***
880 HI = INT(RC/256)
890 LO = RC-HI*256
900 PRINT#1,"P"+CHR$(2)+CHR$(LO)+CHR$(HI)+CHR$(1)
910 RETURN
```

Voraussetzung ist, daß Befehlskanal und REL-Datei bereits geöffnet sind. Die gewünschte Datensatznummer legt man in der Variablen RC ab. Per Unterprogramm wird sie in Low- und High-Byte zerlegt und der Floppy-Schreib-Lesekopf aufs erste Zeichen des selektierten Records gesetzt. Dann kann man mit dem Hauptprogramm per INPUT# (besser: GET#) Daten aus dem Record holen oder mit PRINT# hineinschreiben.

Kleines Programmprojekt gefällig (z.B. eine Adreßdatei)? Auch die idealste relative Dateiverwaltung kommt ohne SEQ-Datei (z.B. mit dem File-Namen »Infodatei«) nicht aus: Darin sind die Datensatznummern der Records und deren Gesamtzahl gespeichert. Nur so ist sichergestellt, daß man beim späteren Laden und Arbeiten mit einer REL-Datei auf alle vorhandenen Datensätze zugreifen kann. Man muß also neben dem Befehlskanal und der REL-Datei auch eine SEQ-Datei einrichten, die dem Hauptprogramm die gespeicherte Anzahl an Adressen übermittelt:

```
100 OPEN 1,8,15
110 OPEN 2,8,2,"INFODATEI,S,R"
120 INPUT #1,F: REM FEHLER AUFGETRETEN?
130 IF F=0 THEN INPUT #2,AD: REM AD=RECORDZAHL
140 CLOSE 2
160 OPEN 2,8,2,"ADRESSEN,L,"+CHR$(100)
```

Zeile 120 zeigt, warum der Befehlskanal manchmal auch als Fehlerkanal bezeichnet wird: Per INPUT# läßt sich jederzeit der Fehlerstatus (Variable F) abfragen. Ist er ungleich 0 (was beim ersten Start unseres Demoprogramms 100prozentig der Fall sein wird, denn das SEQ-File »Infodatei« gibt's noch gar nicht auf Diskette!), ruft das Programm die CLOSE-Anweisung in Zeile 140 auf und geht dazu über, die REL-Datei »Adressen« auf Diskette einzurichten. Die Variable AD bleibt beim Wert 0.

Das Benutzermenü fürs Hauptprogramm soll bei unserem Demobeispiel recht simpel gestaltet sein und die Tasten 1 bis 4 zum Aufruf der entsprechenden Funktionen vorsehen:

```
190 REM **** MENUE *****
200 PRINT: PRINT: "1=EINTRAGEN/2=BLAETTERN/3=
AENDERN/4=ENDE": PRINT
210 GET T$: IF T$="" THEN 210
230 IF T$< >"4" THEN ON VAL(T$) GOSUB 340,430,550:
GOTO 200
250 CLOSE 2
260 PRINT #1,"S:INFODATEI"
270 OPEN 2,8,2,"INFODATEI,S,W"
280 PRINT #2,AD: REM JETZT INFODATEI ERZEUGEN!
290 CLOSE 2: CLOSE 1: REM ALLE DATEIEN SCHLIESSEN
300 END
```

Bei der ersten Arbeit mit der REL-Datei ist der Eintrag gewünschter Adressen unser Hauptanliegen. Dazu soll uns das Unterprogramm »Eintragen« dienen, das weitere Unterroutinen aufruft:

```
330 REM *** NEUE ADRESSEN EINTRAGEN ***
340 PRINT "EINTRAGEN"
350 GOSUB 650
360 AD = AD + 1
370 RC = AD: GOSUB 880
380 GOSUB 730
390 RETURN
640 REM *** ADRESSE ABFRAGEN ***
650 N$="": W$="": T$=""
660 INPUT "NAME:";N$
670 INPUT "WOHNORT";W$
680 INPUT "TELEFON";T$
690 RETURN
```

Jede neue Adresse wird hinter der letzten vorhandenen gespeichert und AD um »1« erhöht (Zeile 360). Die Record-Nummer RC nimmt den aktuellen Wert von AD an und positioniert den Floppyzeiger (Zeile 370). Den Eintrag der neuen Daten (per INPUT-Abfrage eingegeben) übernimmt ein weiteres Unterprogramm:

```
720 REM *** DATENSATZ SCHREIBEN ***
730 IF N$="" THEN N$="*"
740 IF W$="" THEN W$="*"
750 IF T$="" THEN T$="*"
755 REM SCHREIBEN VON LEERSTRINGS VERMEIDEN!
760 PRINT#2, N$ CHR$(13) W$ CHR$(13) T$
770 RETURN
```

Die Zeilen 730 bis 750 verhindern eine kleine Unebenheit bei der Arbeit mit REL-Dateien: Auf gar keinen Fall darf man leere Strings in den Record eintragen! Man muß also vor dem Schreiben unbedingt prüfen, ob die definierten Stringvariablen Zeichen enthalten. Falls nein, muß man ihnen Phantom-Bytes zuweisen (in unserem Beispiel den Stern).

Das größere Problem ist die Vorschrift relativer Dateien, den gesamten Datensatz in einem Rutsch per PRINT#-Befehl im Record zu verewigen. Nach jeder PRINT#-Anweisung positioniert das Laufwerk den internen Floppyzeiger auf den nächsten Datensatz.

Die Lösung zeigt uns Zeile 760. Alle drei Adreßteile (Datenfelder) lassen sich mit einer einzigen PRINT#-Anweisung speichern – wenn man zwischen jedem Eintrag pro Datenfeld ein Carriage Return (= CHR$(13)) einbaut. Ohne dieses Sonderzeichen würde das Laufwerk beim späteren Lesen der Daten (per INPUT#) nicht erkennen, wo der erste Adressenteil (Name) endet und der nächste (Wohnort) beginnt.

Wie in einem Adreßbuch möchten Sie sicher auch in Ihrer Computerdatei blättern: Diese Funktionen überlassen wir den Cursor-Tasten.

- <CRSR rechts>: nächste Recordnummer,
- <CRSR links>: vorhergehende Datensatznummer:

```
420 REM *** BLAETTERN ***
430 PRINT "BLAETTERN"
440 GET A$: IF A$="" THEN 440
445 REM CURSOR RECHTS UND LINKS ABFRAGEN:
450 IF A$< >CHR$(29) AND A$< >CHR$(157)
THEN RETURN
460 IF A$=CHR$(29) AND RC<AD THEN RC=RC+1
470 IF A$=CHR$(157) AND RC>1 THEN RC=RC-1
480 GOSUB 880
490 GOSUB 810
500 PRINT: PRINT N$: PRINT W$: PRINT T$: PRINT
510 GOTO 440
```

Mit jeder anderen Taste außer <CRSR links> und <CRSR rechts> geht's zurück ins Hauptmenü. Trägt man z.B. drei Adressen ein, erhält RC den Wert 3. <CRSR links> vermindert den Variableninhalt um »1«: RC zeigt jetzt auf Datensatz Nr. 2. In Zeile 480 sucht das Programm den Record heraus und liest die eingetragenen Daten per Unterprogrammzeilen ab 810. Zeile 500 bringt den Datensatz auf den Bildschirm. Dann kehrt das Hauptprogramm wieder zum Modul »Blättern« zurück.

Halt, uns fehlt noch die Unterroutine zum korrekten Lesen des Datensatzes:

```
800 REM *** RECORD LESEN ****
810 INPUT#2,N$: IF N$="*" THEN N$=""
820 INPUT#2,W$: IF W$="*" THEN W$=""
830 INPUT#2,T$: IF T$="*" THEN T$=""
840 RETURN
```

Dieser Programmteil ähnelt sehr den Routinen, die wir von der sequentiellen Dateiverwaltung kennen. Dennoch besteht ein kleiner Unterschied: Das Phantomzeichen <*> wird nun wieder in den ursprünglichen Zustand verwandelt (Leerstring). Bei der Bildschirmausgabe sind Leerstrings kein Hindernis! Anderseits wären Sternchen in einer Adreßverwaltung fehl am Platz. Durch Umwandlung in einen Nullstring eliminiert man sie aus dem Datensatz.

Wenn Ihr Programm nur eine Funktion zum Lesen der Datensätze kennt, macht das zwar Sinn, wenn man Adresse oder Telefonnummer z.B. eines guten Freundes vergessen hat: Was aber, wenn er umzieht? Zu einer guten Dateiverwaltung (auch einer REL-Datei) gehört eine Option, bestehende Datensätze zu ändern. Das läßt sich allerdings leicht realisieren: Die noch fehlende Programmfunktion ist (fast) identisch mit dem Eintragen der Adressen. Wichtig ist, daß man die Adresse mit »Blättern« sucht, damit die Variablen AD und RC korrekte Werte erhalten. Der neue Inhalt des Records ersetzt den alten. Geben Sie die neuen Daten ein, sie werden Byte für Byte an dafür vorgesehener Stelle plaziert:

```
540 REM *** RECORD AENDERN ***
550 PRINT "AENDERN"
560 GOSUB 650
570 GOSUB 880
580 GOSUB 730
590 RETURN
```

Zeile 560 fragt nach der neuen Adresse (wie bei »Adresse eintragen«), dann wird der Zeiger neu positioniert (das Programm übernimmt die aktuelle Record-Nummer, die sich beim Blättern präsentiert). Zeile 580 ruft schließlich die Unterroutine »Datensatz schreiben« auf. Jetzt wird der alte Inhalt des Datensatzes endgültig durch den neuen ersetzt.

Unser Demoprogramm ist modulartig aufgebaut und so konstruiert, daß es jederzeit von Ihnen für eigene Zwecke verändert und erweitert werden kann (professionellere Bildschirmausgabe, größere Datenmenge pro Record mit weiteren Datenfeldern, z.B. Straße, Beruf, Geburtstag usw. Nicht vergessen: Die Summe aller Einträge zu den einzelnen Datenfeldern darf 254 Byte nicht überschreiten!). Außerdem ist man keineswegs an die typischen Datenfeldbezeichnungen von Adreßdateien gebunden: Im Programmodul ab Zeile 640 lassen sich beliebige Abfragekriterien (z.B. für eine Videodatei) eintragen. Wenn Sie noch eine Unterroutine zur Druckausgabe ins Programm einfügen, ist der Computer-Karteikasten perfekt! (bl)

Rückholaktion – auch das geht

Schnell ist's passiert: Ein wichtiges Programm, eine Textdatei oder eine arbeitsintensive Grafik werden durch einen unvorsichtigen Befehl von Diskette gelöscht! Vor allem, wenn man die SCRATCH-Anweisung der Floppy mit dem Jokerzeichen <*> verwendet, tilgt man oft ungewollt Dateien von Disk, die gar nicht gemeint waren!

Kein Problem, wenn's Ihnen sofort auffällt und die Disk nicht mehr zur Datenspeicherung verwendet wird: Die Inhalte der Datensektoren sind nämlich gar nicht gelöscht, sondern nur der Dateityp! Er erhält bei SCRATCH die Kennzeichnung $80 (128) – für die Floppy bedeutet dieser Wert, daß es sich um ein DELeted-File handelt! Allerdings werden die Nummern der belegten Sektoren im Blockbelegungplan wieder freigegeben – sie stehen also zur Speicherung beliebiger Daten frei und können (müssen aber nicht) bei einem folgenden SAVE-Vorgang wieder mit neuen Daten beschrieben werden.

Vorausgesetzt, man hat anschließend nichts mehr auf Diskette gespeichert, kann man z.B. per Diskettenmonitor den Dateityp korrigieren (statt $80 = DEL Typ $81 = PRG eintragen). Ruft man nach einem VALIDATE nun erneut das Directory auf, stellt man erfreut fest: Das Programm ist wieder da!

Komfortabler geht's mit unserem Utility:

```
LOAD "FILE REVIVAL",8
```

Nach dem Start mit RUN wird man aufgefordert, die entsprechende Disk einzulegen und <RETURN> zu drücken.

```
FILE REVIVAL
fuer Floppy 1541/1570/1571

stellt mit SCRATCH geloeschte
Dateien auf Diskette im
Originalzustand wieder her.
ACHTUNG! Es duerfen vorher
keine neuen Dateien auf die
Diskette gespeichert werden!

Disk einlegen, dann <RETURN>

DEL iceball

Sektoren wiederverbinden? (j/n)
```

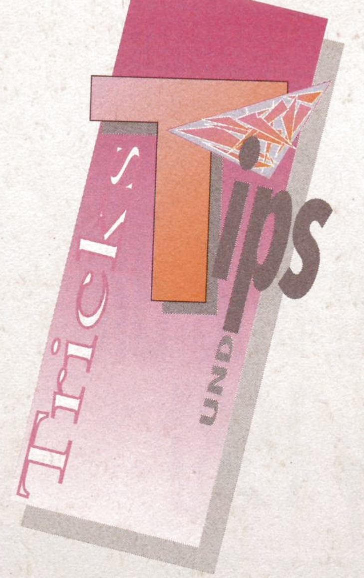
[3] File Revival: Sie entscheiden, ob die Datei zurückgeholt werden soll!

Die Floppy startet, durchforstet das Directory nach DEL-Dateien und zeigt die jeweiligen Files einzeln in einem Window. Wird das Laufwerk fündig, unterbricht es die Suche mit der Frage »Sektoren wieder verbinden? (j/n)« (Abb. 3). Bei <J> erscheint die Meldung »File wiederherstellen« (das dauert seine Zeit, je nach Umfang der gelöschten Datei). Dann müssen Sie den Anfangsbuchstaben des entsprechenden File-Typs angeben:

- <P>RG: Programm (Floppy-Kenn-Byte im Directory: $81),
- <S>EQ: sequentielle Datei ($82),
- <U>SR: User-File ($83),
- <R>EL: relative Datei ($84).

Nach der Meldung »x Blocks wieder belegt« macht das Programm mit der Suche nach anderen gelöschten Dateien im Directory weiter. Anschließend aktiviert das Programm die Floppy-Anweisung VALIDATE (das dauert je nach Directory-Umfang ebenfalls einige Sekunden!).

Das Programm fragt jetzt, ob Sie eine andere Diskette nach verschollenen Dateien durchforsten wollen, um sie wiederzubeleben (<J>) oder das Programm ohne Reset beenden möchten (<N>). (bl)



Hin ist hin:
Dateien für immer gelöscht!

Bei diesem Utility hätte »File Revival« keine Chance: »Physical Scratch« vernichtet nicht nur die File-Typ-Bezeichnung, sondern füllt alle zur Datei gehörenden Sektoren mit Nullbytes auf. Alle Wiederbelebungsversuche sind zum Scheitern verurteilt – auch per Diskettenmonitor!

Damit nimmt man Datenschnüfflern jede Chance (wenn Sie z.B. Bekannten eine Diskette schicken, auf der sich gelöschte Dateien befinden). Mit »File Revival« wär's leicht, zumindest bestimmte Files zu reanimieren – ganz gegen Ihre Absichten!

Das vermeiden Sie nachhaltig mit unserem Utility:

```
LOAD "PHYSICAL SCRATCH",8
```

Nach dem Start mit RUN müssen Sie den gewünschten Dateinamen eingeben (korrekt ausgeschrieben, das Jokerzeichen <*> nützt nichts!). Am besten drucken Sie vorher das Diskinhaltsverzeichnis aus – das Programm besitzt keine Directory-Funktion!

Für alle Eventualitäten folgt eine Sicherheitsabfrage, mit der Sie die Wahl rückgängig machen (<N>) oder per Taste <Q> ganz aus dem Programm aussteigen können (Abb. 4). Mit <Y> wird die Löschaktion eingeleitet – die Datei ist unwiederbringlich futsch!

Dann springt das Programm erneut zur Namenseingabe fürs nächste File, das Sie löschen möchten. Da es keine Ende-Funktion gibt, verläßt man das Utility per <RUN/STOP RESTORE>.

```
PHYSICAL SCRATCHER V1.0 BY I.HENDRICKS
ENTER FILE NAME:ICEBALL
ARE YOU SURE? (Y/N/Q)
```

[4] Physical Scratch: Vorsicht! Außer den Blockverbindern bleibt von den Datensektoren nichts übrig.

Wenn Sie nun einen Diskettenmonitor laden, wird's offensichtlich: Das Programm hat den File-Typ auf $00 gesetzt (nicht auf $80 (128 dez.)), wie's das Floppy-DOS üblicherweise bei gelöschten Files macht. Damit wäre die Dateikennung mit jedem Unscratch-Programm (z.B. »File Revival« auf der beiliegenden Diskette zum 64er-Sonderheft 83) leicht zu identifizieren. Die Blockverbinder pro Sektor wurden zwar nicht gelöscht (sie lassen sich per Diskmonitor leicht verfolgen). Wenn Sie aber die Inhalte als Hexdump auf den Bildschirm bringen, lächeln Ihnen nur Null-Bytes entgegen!

(I. Hendricks/bl)

128er Corner

CP/M 3.0 und die Floppy 1581

Achtung, C-128-Besitzer mit der CP/M-Version vom 28. 5. 87: Sie arbeitet problemlos mit der 1581 zusammen!

Nutzen Sie auch unter CP/M 3.0 den immensen Speicherplatz einer 3½-Zoll-Disk! Das Dienstprogramm FORMAT.COM der neueren CP/M-Fassung akzeptiert das 1581-Laufwerk.

– Booten Sie CP/M 3.0 von der 5¼-Systemdisk im Laufwerk Nr. 8 (1541 oder 1571), die 1581 sollte als Gerät Nr. 9 eingestellt sein.

– Holen Sie den CP/M-Formatter in den Speicher:

```
A>format.com
```

– Wählen Sie den Menüpunkt »Select Different Drive« und und stellen Sie »B« ein. CP/M hat nun die 1581 als Zweitfloppy akzeptiert und stellt die vorgesehene Option ein: »Commodore 1581 (Double Sided)« (Abb.).

– Legen eine 3½-Zoll-Diskette in die 1581 (Gerät Nr. 9, Drive B) und drücken Sie <$>. Die Disk wird formatiert und die CP/M-Systemspur inkl. Boot-Sektor eingetragen.

Jetzt steht die Micro-Disk als vollwertige Arbeitsdiskette unter CP/M zur Verfügung. Wer daraus eine bootfähige Systemdisk machen will, muß das CP/M-Kopierprogramm PIP.COM verwenden:

```
A>pip b:=a:*.*
```

Alle Dateien der 5¼-Zoll-Systemdisk (inkl. CPM+.SYS und CCP.COM) werden nun auf die 3½-Zoll-Scheibe übertragen: die CP/M-3.0-Micro-Disk ist fertig!

Beim Starten gibt's allerdings Probleme mit der Geräteadresse: Wenn die Floppy 1581 weiterhin Nr. 9 behält, nützt auch der C-128-Befehl »Boot U9« nichts. Sie muß unbedingt auf Gerätenummer 8 eingestellt werden. Wer einen C 128D mit integrierter Floppy 1571 (Geräteadresse stets 8) besitzt, kann das interne Laufwerk mit einer simplen Burst-Anweisung umstellen:

```
open1,8,15,"u0>"+chr$(9): close1
```

Jetzt läßt sich CP/M 3.0 von der Sicherheitskopie auf der 3½-Zoll-Disk im Laufwerk 1581 (per DIP-Schalter auf Nr. 8 eingestellt) problemlos starten. (bl)

```
C128 CP/M FORMAT PROGRAM
     08 JUNE 1987
Drive B is a 1581



Please select disk type to format
Commodore 1581 (Double Sided
Write system track only
Select different Drive
```

FORMAT.COM (Version 8.6.87) identifiziert die Floppy 1581 als Zweitlaufwerk



Exakte Bauanleitungen vom Wochenendprojekt bis zur komplexen Anwendung – natürlich sowohl mit getestetem Platinenlayout als auch mit der entsprechenden Software.

- Durch die Umbauanleitung »Laptop 64« sind C64 und Floppy endlich kompakt in einem Gehäuse untergebracht und netzunabhängig.
- Ein Universalmeßplatz bietet Raum für bis zu acht Module.
- »EXOS« beschleunigt Ihre Floppy und bietet viele Befehle. Wir liefern gleich die Anleitung für EPROM-Programmiergerät und Adaptersockel mit.
- Dem C64 verhelfen wir zum guten Kontakt zur Umwelt. Unter anderem mit einer unkomplizierten Schaltung zur Steuerung von bis zu acht Verbrauchern.
- Sound-Freaks erhalten hier Schaltung, Layout und Testsoftware für ein MIDI-Interface.
- Last but not least ist auch für die Dunkelkammer-Freaks etwas dabei: Ein universeller Foto-Timer.

Nr. 84 gibt's ab 20.11.92 bei Ihrem Zeitschriftenhändler

Aus aktuellen oder technischen Gründen können Themen ausgetauscht werden. Wir bitten dafür um Verständnis.

7/92

FILTER CIGARETTES

Marlboro

QUALITY CIGARETTES

Marlboro

Die EG-Gesundheitsminister: Rauchen gefährdet die Gesundheit. Der Rauch einer Zigarette dieser Marke enthält 0,9 mg Nikotin und 13 mg Kondensat (Teer). (Durchschnittswerte nach ISO)



✌ *PowerPlay bringt Euch jeden Monat die stärksten, heissesten und aktuellsten Spiele ins Haus! Getestet und gnadenlos beurteilt.*

✌ *PowerPlay prüft für Euch alles, was an Spiele-Soft- und Hardware auf den Markt kommt! Gründlich und absolut objektiv.*

✌ *PowerPlay hilft Euch mit Tips&Tricks bei den schwierigsten Spielen! Profimäßig und super clever.*

PowerPlay - die ganze Welt der Spiele in einem Heft! Holt Euch jetzt die neueste Ausgabe! Es lohnt sich...